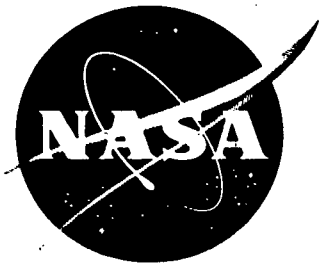
NASA TECHNICAL MEMORANDUM

NASA TM X-58065
November 1971



AN ANALYSIS AND COMPARISON OF SEVERAL TRAJECTORY OPTIMIZATION METHODS

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MANNED SPACECRAFT CENTER

HOUSTON, TEXAS 77058

NASA TM X-58065

AN ANALYSIS AND COMPARISON OF SEVERAL TRAJECTORY OPTIMIZATION METHODS

J. M. Lewallen
Manned Spacecraft Center
Houston, Texas

PREFACE

The optimization and control of spacecraft trajectories has been of considerable interest during the past decade, and a significant amount of progress has been made in developing a theoretical and numerical capability to solve complex trajectory problems. There still exists, however, a need to determine the best approach, given a specific problem. The generality of such a task is overwhelming, but an initial step is taken when most of the promising methods have been studied with the aid of a specific, but representative example. This dissertation takes this first step, and along with several significant theoretical and numerical contributions, compares the relative merits of several trajectory optimization methods.

In each stage of this research, the author has benefited from many valuable suggestions by and discussions with many individuals. He wishes to express sincere appreciation to W. T. Fowler, G. J. Lastman, and J. F. Jordan who, as fellow students, provided considerable encouragement. He wishes to express gratitude to Professors L. Clark, W. Carter and E. Prouse of The University of Texas for reading the manuscript and making helpful suggestions. The author is especially indebted to R. D. Witty of the Lockheed Electronic Corporation, without whose patience, intelligence and persistence the endeavor, as presented, would have never been realized.

The author also wishes to express gratitude to E. L. Davis, Jr. of the Manned Spacecraft Center for providing a stimulating environment and great encouragement during the course of this study. He is especially indebted to Professor B. D. Tapley of The University of Texas who served as advisor, but more than that a good friend and teacher and a constant source of inspiration and guidance.

The author wishes to express his deepest gratitude to his wife and children, whose love and patience made the whole undertaking worth doing.

Houston, Texas
May 1966

Jay M. Lewallen

ABSTRACT

A theoretical development and comparative evaluation is made for several methods of solving the problem associated with the optimum transfer of a spacecraft. Particular attention is given to the sensitivity of the convergence characteristics of the methods to initially assumed parameters and trial solutions, convergence times, computer logic and storage requirements.

The methods considered may be classified as one of the following types: (1) Perturbation, Second Variation or Extremal Field Methods, (2) Quasilinearization or Generalized Newton-Raphson Methods, or (3) Gradient or Steepest Descent Methods. The numerical comparison of the convergence characteristics is made by considering a minimum time, low thrust, Earth-Mars transfer trajectory.

A new quasilinearization method, called the Modified Quasilinearization Method, is proposed. For the example considered, this method reduces convergence time by approximately 70% when compared with the Generalized Newton-Raphson Method. Moreover, the method allows the terminal boundary to be specified by a general function of the problem variables rather than individual values of the variables themselves.

A uniquely specified and easily determined, time dependent weighting matrix has been discovered for the gradient techniques. This weighting matrix accelerates the shaping of

the optimal control program and improves the convergence characteristics during the terminal iterations by giving more weight to regions of low sensitivity.

Convergence envelopes, which give an indication of how sensitive the convergence characteristics are to initially assumed parameters, are plotted for the Perturbation and Quasilinearization Methods. Several iteration schemes are proposed which significantly increase the size of the convergence envelopes, and hence decrease the sensitivity of the method to initially assumed parameters.

TABLE OF CONTENTS

TABLE OF CONTENTS

(CONT'D)

TABLE OF CONTENTS

(CONT'D)

LIST OF FIGURES

LIST OF SYMBOLS

The following list tabulates all significant symbols used in the main text. Each symbol is accompanied by a brief description and the equation number where the symbol is first introduced. A definition of each symbol is given where the symbol is introduced.

Matrices:

The matrix size is indicated in the statement immediately following the symbol. The following specific indices are used.

- n - the number of state variables
- m - the number of control variables
- p - the number of initially specified constraint relations
- q - the number of terminally specified constraint relations

A — 2n x 2n matrix of partial derivatives, (3.2)

F — n x n matrix of partial derivatives, (5.5)

G — n x m matrix of partial derivatives, (5.5)

I — 2n x 2n unity matrix, (3.10)

W — m x m matrix of arbitrary weighting terms, (5.19)

LIST OF SYMBOLS

(CONT'D)

Y	2n x 2n-p matrix of homogeneous solutions, (4.6)
${}_1\Theta$	2n x 2n matrix of partial derivatives resulting from the 2n backward integrations of the 2n vector of adjoint equations, (3.10)
${}_2\Theta$	2n x 2n matrix of partial derivatives resulting from the 2n forward integrations of the 2n vector of adjoint equations, (3.14)
Θ'	n+1 x 2n matrix of partial derivatives resulting from the n+1 backward integrations of the 2n vector of adjoint equations, (4.15)
Θ	n+1 x 2n matrix of partial derivatives resulting from the n+1 backward integrations of the 2n vector of adjoint equations, (3.17)
${}_1\Phi$	2n x 2n matrix of partial derivatives resulting from the 2n forward integrations of the 2n vector of perturbation equations, (3.22)
${}_2\Phi$	n+1 x 2n matrix of partial derivatives resulting from the n+1 forward integrations of the 2n vector of perturbation equations, (3.27)
Φ	2n x n matrix of partial derivatives resulting from the n forward integrations of the 2n vector of perturbation equations, (3.29)

Vectors:

All vectors are column vectors unless otherwise noted. The vector size is indicated in the statement immediately following the symbol, where the indices are defined in the previous section on matrices.

LIST OF SYMBOLS

(CONT'D)

B	2n vector of nonhomogeneous terms, (4.3)
c	n+1 vector of desired percentage corrections in the terminal constraints, (3.31)
C	2n-p vector of corrections required for the assumed initial conditions, (4.6)
f	n vector of state variable derivatives, (2.1)
F	2n vector of state and Euler-Lagrange variable derivatives, (2.37)
g	n vector of initial constraint relations, (2.40)
h	n+1 or q+n+1 vector of terminal constraint relations, (2.32)
dh	n+1 or q+n+1 vector of terminal dissatisfaction change, (3.4)
u	m vector of control variables, (2.1)
w	2n vector of nonhomogeneous solutions (4.5)
x	n vector of state variables, (2.1)
y	2n vector of homogeneous solutions, (4.4)
z	2n vector of state variables and Euler-Lagrange variables, (2.37)
$d\beta$	q+1 vector defined as $d\Psi - (\lambda^T_{\Psi\Omega}\delta x)_0$, (5.29)
η	p vector of specified initial constraint relations, (2.3)
λ	n vector of time dependent Lagrange multipliers, (2.5)
Λ	2n vector of adjoint variables, (3.3)
ν	q vector of constant Lagrange multipliers, (2.5)
Ψ	q vector of specified terminal constraint relations, (2.4)

LIST OF SYMBOLS

(CONT'D)

Scalars:

a	constant that represents the terminal value of time, (4.9)
E	Weierstrass E-Function, (5.50)
GM	universal gravitational constant, (A.1.1)
H	generalized Hamiltonian equal to $\lambda^T f$, (2.6)
I	auxiliary functional to be extremized, (2.5)
K	step size in gradient space, (5.46)
m	instantaneous spacecraft mass, (A.1.1)
P	auxiliary functional equal to $\phi+\nu^T\Psi$, and penalty function, (5.35)
r	instantaneous radial position, (A.1.1)
s	independent variable where $t = as$, (4.9)
dS	step size in control space, (5.19)
t	independent variable, time, (2.1)
T	thrust, (A.1.1)
u	instantaneous radial velocity, (A.1.1)
v	instantaneous tangential velocity, (A.1.1)
W_i	weighting constants where $i = 0,q$, (5.35)
μ	constant Lagrange multiplier, (5.24)
ρ	metric defined in Eq. (4.7)
ϕ	performance index, (2.2)
Ω	stopping condition, (5.4)

LIST OF SYMBOLS

(CONT'D)

Superscripts:

$(\dot{})$	differentiation with respect to time
$(\ )^T$	matrix transpose
$(\ )'$	first variation or first derivative with respect to s
$(\ )''$	second variation
$(\ )^{-1}$	matrix inverse
$(\ )^*$	refers to an optimal trajectory
$(\bar{})$	refers to an assumed value

Subscripts:

$(\ )_0$	specified initial value
$(\ )_f$	specified terminal value
$(\ )_{\lambda,u,x,t}$	partial derivative of the subscripted quantity with respect to λ, u, x, t, respectively
$(\ )_n$	n^{th} trajectory iteration
$(\ )_k$	k^{th} time iteration
$(\ )_e$	Earth
$(\ )_m$	Mars
$(\ )_s$	Sun

Miscellaneous Symbols:

d	total variation or differential operator, i.e. $d(\ ) = \delta(\ ) + (\dot{})\, dt$

LIST OF SYMBOLS

(CONT'D)

δ	variation operator
$X(t_f)$	evaluation of the instantaneous variable X at time t_f

Abbreviations:

MAF	Method of Adjoint Functions
MPF	Method of Perturbation Functions
MGNR	Method of Generalized Newton-Raphson
MMGNR	Modified Method of Generalized Newton-Raphson
MQM	Modified Quasilinearization Method
MSD	Method of Steepest Descent
MMSD	Modified Method of Steepest Descent
AU	Astronomical Unit

CHAPTER I

INTRODUCTION

A treatise on the theory of trajectory optimization and its application requires a clear and meaningful definition of the problem. This definition should include a discussion of the terms and concepts required in studying the background material and the theoretical formulations. An indication of the purpose of the investigation is given along with the extent or scope of such a study.

1.1 Definition of the Optimization Problem

The optimization of spacecraft trajectories has been of considerable interest for a number of years, and significant progress has been made in developing a capability for solving very complex trajectory problems. In one class of optimization problems, it is desired to determine the history of the control variables in such a manner that certain specified initial and terminal constraints are satisfied while some performance index is extremized. The control variables are unspecified inputs to the system which may be chosen to control the state, i.e., the position and velocity. The initial and terminal constraints are simply conditions on the position and velocity that must be satisfied at the initial and terminal time, respectively. The performance index is usually a scalar function associated with the spacecraft performance and is the quantity to be extremized. It

may be a scalar function of the terminal state and time and/or a scalar integral term evaluated along the trajectory.

The calculus of variations is the classical tool for solving such problems, and with its use necessary conditions for an optimal trajectory may be derived. These necessary conditions are derived in Chapter 2 and consist of boundary conditions referred to as transversality conditions, algebraic equations referred to as optimality conditions and the Euler-Lagrange differential equations. The optimality conditions and the Euler-Lagrange equations must be satisfied at each point in the time interval of interest. A closed form solution for these equations and boundary conditions is very difficult to obtain and has been obtained for only a few relatively simple cases. When an optimization problem is solved numerically in such a way that the necessary conditions are satisfied, the method is usually designated an indirect method.

There have been alternate methods developed to solve the above stated class of problems without using the necessary conditions derived with the calculus of variations. These methods, usually referred to as direct methods, use influence functions which indicate how the performance index and terminal constraints are influenced by initial state variations and integrated control variations.

In both the indirect and direct methods, the terminal constraints are handled in either the so-called "hard" or "soft" forms. In the "hard" form an effort is made to satisfy the terminal constraints identically while in the "soft" form the

terminal constraints are satisfied only approximately. It is with this latter case that the penalty function concept to be discussed later is introduced. The philosophy used in this method is that a certain penalty is accepted because of the approximate satisfaction of the terminal constraints.

1.2 Background Study of Optimization Theory

In assessing the "state of the art" in trajectory optimization theory and application, it is helpful to understand the developments that lead to this current state. This background is divided into previous and recent developments, the recent developments being made since about 1960. The distinction between indirect and direct methods has become increasingly clear during these recent years and are discussed separately.

1.2.1 Previous Developments

The original trajectory optimization problems were formulated in terms of a set of nonlinear, ordinary differential equations, which were required to satisfy split boundary conditions. The first problems to be solved were extremely simple since numerical solution of the more difficult problems required extensive computations. With the advent of the high speed digital computer, several previously impractical methods became available for numerical solutions. Development of the computer has stimulated the formulation of many previously unknown methods.

Some of the first published formulations of optimal trajectory programming problems appeared in the early 1950's. One

of the best known was by Lawden (1)* in which the equations which described the optimal trajectory were derived for the general case of a rocket moving in a specified gravitational field and subject to atmospheric resistance. However, results for only the highly specialized case of uniform gravitational field and no atmospheric resistance are presented. The analysis probably represents one of the most difficult known cases for which a closed form solution can be obtained.

In August 1957, a classical paper was published by Breakwell (2) in which a method was presented for using a high speed digital computer for the study of a broad class of trajectory optimization problems. This class includes boost trajectories for maximum range or maximum energy, minimum time intercept trajectories, and maximum glide range trajectories. The method devised for determining a solution requires a guess for unknown initial conditions and an interpolation procedure to decrease the terminal constraint dissatisfaction on each successive iteration. This particular approach can become extremely time-consuming and inefficient.

A different analytical development of trajectory optimization theory was published by Kelley (3) in October 1960. The method is referred to as the gradient method and it is based on an extension of some ideas presented by Courant in 1941. The gradient technique represented a completely different approach

---

*Numbers appearing in parenthesis following a name refer to publications listed in the References.

to the solution of optimization problems, and it soon became evident that the recently developed optimization schemes would fit into two basically different classifications, the indirect and direct trajectory optimization methods.

The indirect methods involve the simultaneous solution of the differential equations of motion and the Euler-Lagrange equations while satisfying at each point in time a local optimality condition. Hence, every trajectory iteration is an optimal trajectory, from the initial to some terminal point in space. The only remaining problem is to satisfy the terminal constraint relations. This approach also includes methods where the differential equations mentioned above are linearized about the previous trajectory iteration, even though the trajectories are not exactly optimal in this case.

The direct methods involve the solution of the differential equations of motion and produce control variable modifications that extremize the desired performance index while decreasing the terminal constraint dissatisfaction. This approach includes the gradient techniques.

1.2.2 Recent Developments

Since 1960 there have been a number of significant improvements for both the indirect and direct trajectory optimization methods. During this recent period a distinct difference between the two approaches has evolved and for this reason the approaches are discussed separately.

1.2.2.1 Indirect Approaches

As mentioned earlier, the capability for solving optimum trajectory problems has existed since the development of the theory to solve the two-point boundary value problem, however, numerical computation schemes were lacking. One of the first recent schemes was published by MacKay, Rossa, and Zimmerman (4) in 1961. The analysis uses a set of differential equations which describe the optimal thrust direction and a criterion for determining the best time at which to begin and end a coast phase. An iteration method is used to solve the two-point boundary value problem. The various partial derivatives that describe how the terminal state changes as the initial state is changed, are evaluated by a first-order finite difference technique and the successive integration of the differential equations.

Melbourne, Sauer, and Richardson (5), also in 1961, presented the results of an investigation of optimum rendezvous and round trip trajectories for a typical mission to Mars. A classical calculus of variation approach is used and a Newton-Raphson technique is implemented for the solution of the two-point boundary value problem. The technique for determining the partial derivative matrix is similar to that used by MacKay, Rossa, and Zimmerman (4) and the suggestion is made that this matrix be updated only once every several trajectory iterations.

The Newton-Raphson optimization method is discussed further by Scharmack (6) and several examples are presented. An especially simple special form of the Newton-Raphson method is

given also for the case where the terminal boundary is a function of time alone.

In 1962 Jurovics and McIntyre (7) presented a method for the systematic evaluation of the two-point boundary value problem using the equations adjoint to the linearized differential equations of motion and the Euler-Lagrange equations. The foundation of this work was laid by Goodman and Lance (8), but the applicability of the technique to systems of nonlinear equations is very limited and the terminal time must be known. Jurovics and McIntyre eliminated some of the restrictions and extended the technique to allow for variable terminal time.

An extension was made to the Newton-Raphson techniques by Breakwell, Speyer, and Bryson (9) in 1963. The procedure is based partially on previous work by Breakwell (10) in 1959. The method uses a set of equations obtained by perturbing the previous nominal trajectory to evaluate the required partial derivative matrix. The generality of the formulation allows for variable terminal time and the satisfaction of time and state dependent terminal constraints. After the partial derivative matrix has been determined, a multiple linear interpolation is made to determine the corrections required for the initial conditions. The Euler-Lagrange equations are satisfied on every iteration, and hence every trajectory is an optimal one. However, the terminal constraints must be satisfied by an iterative process.

A rather recent development based on the theory of the second variation was published by Kelley, Kopp, and Moyer (11)

in 1963. In the initial phase of computation, the penalty function concept of handling the terminal constraints is used, and the process behaves much like the classical gradient technique. During the terminal phase, the constraints are satisfied exactly and the method converges more rapidly than the gradient scheme. However, the second variation method is significantly more complicated, theoretically and computationally, than the first order gradient theory. However, the reference does state that this disadvantage is partially offset by a reduction in required computational time.

Jazwinski (12) in 1964 presented an extension to the method suggested by Jurovics and McIntyre (7) by using the adjoint system to solve optimization problems which contain initial and terminal boundary conditions that are general functions of the problem variables. An additional feature of this scheme is that after the open-loop optimization problem has been solved all the information for the closed-loop control problem is available. This information is also available in Breakwell, Speyer, and Bryson's (9) paper, but it must be pointed out that Jazwinski's method requires fewer integrations of an equivalent set of equations.

A different approach to the solution of the indirect optimization problem has been suggested by McGill and Kenneth (13) in 1964. This method, called the Generalized Newton-Raphson Method, is formulated through the use of the quasilinearization concept as presented by Kalaba (14). A convergence proof for the method was presented by McGill and Kenneth (15) in 1963. This

method uses the linearized versions of the differential equations of motion and the Euler-Lagrange equations, and proceeds to solve a sequence of linear problems, the solutions of which converge to the solution of the desired nonlinear problem. A set of perturbation or homogeneous equations are used to determine the partial derivative matrix. The implementation of the procedure is similar to the perturbation method presented by Breakwell, Speyer, and Bryson (9). The method is distinguished by the fact that an initial solution must be assumed rather than just the initial values of the dependent variables. Furthermore, variable terminal time problems are handled in a very awkward manner.

The awkward handling of terminal time is partially reduced by Long (16) by introducing a change in the independent variable. The method proposed by Long is still rather cumbersome because an additional differential equation must be integrated and all the previous equations are complicated by another complex term. It is shown, however, by McGill and Kenneth (15), that if convergence does occur it does so quadratically, and that the terminal constraints, which are not general functions of the problem variables, can be identically satisfied on every trajectory iteration.

In summary, the indirect optimization methods are usually formulated in terms of a two-point boundary value problem, and hence the many methods previously used for solution of this type of problem become applicable for the solution of trajectory optimization problems. One of the most significant advantages of the indirect methods is that the convergence properties are excellent.

Another advantage is that the converged solution does represent a true optimal, not just an approximation. The most severe disadvantage is that the solution of the differential equations is highly sensitive to the initially assumed values of the dependent variables. This implies that accurate initial values are needed to start the integration, and the problem is compounded by the fact that often little physical significance can be attached to the initial values of the Euler variables.

The disadvantages associated with indirect optimization methods are severe enough to encourage the forumulation of methods that eliminate these difficulties. The convergence of the direct optimization methods are not as dependent on the initially assumed parameters as are the indirect methods, but some extremely undesirable characteristics are introduced. A brief discussion of the direct methods is given in the following section.

1.2.2.2 Direct Approaches

While the gradient theory for flight path optimization was being developed by Kelley (3), a similar formulation was being made simultaneously and independently by Bryson, Denham, Carroll, and Mikami (17) (18). In Reference (17), the gradient method is used to study the problem of determining a control variable program that minimizes vehicle heating during reentry to the earth's atmosphere.

In 1961, Kelley, Kopp, and Moyer (19) presented an analysis of several gradient methods using inequality constraints on the control variables and a penalty function technique for

handling terminal constraints. It is pointed out in the study that the numerical results obtained were too limited for comparing the relative merits of the methods.

In an effort to determine the thrust steering program for the optimization of a second stage booster, Pfeiffer (20) developed a method of "critical direction" which was similar to the gradient techniques of Kelley and Bryson. This same gradient concept is studied by Wagner and Jazwinski (21) and both terminal and instantaneous inequality constraints are introduced into the formulation. Wagner and Jazwinski also present an interesting method for determining the step size magnitude that should be taken in the gradient direction to approximately maximize the decrease in the penalty function.

The gradient technique is well defined and has been quite successful in avoiding the difficulties associated with the two-point boundary value problem associated with the calculus of variation necessary conditions. One of the most costly deficiencies of this method is the poor convergence characteristics in the terminal stage of convergence. In 1963, Rosenbaum (22) developed a method similar to a closed-loop guidance scheme that provides rapid convergence for a variety of missions. The distinctive feature of this method is that the step size in the gradient direction is calculated and becomes a time dependent quantity. The significant result is that larger deviations from the nominal trajectory can be tolerated while still satisfying the terminal constraints, thus it is possible to move more rapidly toward the optimal trajectory.

Stancil (23), in 1964, presented a slightly different approach to the inherent gradient convergence problem. This approach is similar to Rosenbaum (22) in that a time dependent weighting matrix is calculated. Basically the formulation followed a suggestion made, but not used, by Bryson, Denham, Carroll, and Mikami (17), in which the current control program was averaged with the Eulerian control.

The latest innovation to an optimization method is reported by McReynolds and Bryson (24), and is called a successive sweep method. To this author's knowledge, no computational results have been published. The procedure represents an extension and unification of the steepest-descent and second variation techniques. The procedure requires the backwards integration of a set of equations, in addition to the usual adjoint equations, that generate a linear control law that preserves the gradient history on the following step. The gradient history, however, may be changed by specified amounts while also specifying a change in the terminal constraint dissatisfaction. Thus, in a finite number of steps, the gradient history and the terminal dissatisfaction can be forced to approach zero. Actually, the method has characteristics similar to indirect methods as well as direct methods.

The method seems very promising from a theoretical point of view, but before a judgment on its applicability to solving trajectory optimization problems can be made, some computational experience must be obtained.

In summary, the direct optimization methods suffer from

poor convergence characteristics, as the optimal trajectory is approached and, in fact, never yields a solution which will satisfy the classical optimality conditions. The methods, however, do begin the convergence process with a relatively poor initial estimate of the control variable history, and seek weak relative extremals as opposed to points where the functional is merely stationary.

1.2.3 Recent Comparisons

The number of published studies that compare the relative merits of the recently developed trajectory optimization schemes is extremely limited. The reason for this is certainly not because this type of knowledge is unwanted or meaningless, but because it is so difficult to select a reasonable basis for comparison. Another discouraging fact is that most optimization methods are highly problem dependent.

One study of three related successive approximation gradient schemes by Kelley, Kopp, and Moyer (19) in 1961 concluded that the numerical results were too limited to provide a comparison of the relative merits. The differences in convergence speeds were insignificant in comparison to the improvements attainable by small adjustments in the penalty function constraints.

A more recent publication by Kopp and McGill (25) and Moyer and Pinkham (26) compares a gradient, second variation and generalized Newton-Raphson technique on both theoretical and computational basis. The theory is explained by considering an

ordinary minimum problem with a side constraint. It is stated in this reference that the second variation method is a specific approach to the generalized Newton-Raphson method. One conclusion made on convergence times is that the second variation scheme requires approximately 50% less computer time than the conventional gradient technique, and the generalized Newton-Raphson method required even less time.

1.3 Purpose of the Investigation

The ultimate purpose of this investigation is to develop an insight into the available numerical optimization methods, so that, given a problem and a set of circumstances, an intelligent choice may be made as to which procedure is best suited for that particular problem. This ultimate purpose is approached by satisfying the following secondary objectives:

(1) Increase the understanding of the currently popular optimization methods so that the deficient areas of each method are discovered. Extend and modify these methods to eliminate the deficiencies.

(2) Formulate a basis on which the methods may be compared, and make a meaningful comparison of the relative merits of each method.

1.4 Scope of the Investigation

The scope of the investigation includes the theoretical development of both direct and indirect methods. These methods

are formulated in the "open loop" form; i.e., information is not fed back to the system to provide control for the inevitable state variations discovered during the process.

The problem is formulated in a Mayer form, and here the performance index is simply a scalar function of the terminal state and terminal time. The terminal constraints, which are of the equality form, may be general functions of the problem variables, and the terminal time may be unknown.

The methods are applied to the study of a two-dimensional transfer trajectory from Earth to Mars. One control variable, the thrust attitude angle, is used. The specified terminal constraints do not contain the time explicited.

CHAPTER 2

FORMULATION OF THE OPTIMIZATION PROBLEM

The theoretical development of several trajectory optimization methods is made with an objective being the presentation of a unified or common approach. A fundamental factor in describing the formulation of any trajectory optimization problem is the derivation of the first necessary conditions for an optimal trajectory, with the appropriate remarks concerning sufficiency. One other requirement helpful to the discussions presented, especially for the indirect optimization development, is an explanation of how the optimization problem is reduced to a two-point boundary value problem.

2.1 Derivation of the Necessary Conditions for an Optimal Trajectory

The classical trajectory optimization problems require that certain necessary conditions be satisfied. The different optimization techniques that have been developed tend to satisfy these conditions in various ways. The necessary conditions are derived from the consideration of the following problem. Determine the history of the variables that control a nonlinear system in such a manner that some index of performance is extremized while certain specified initial and

terminal constraints are satisfied. This performance index is usually some function of the terminal state and time.

The differential equations of motion that describe the trajectory of a spacecraft may be derived by applying Newton's Second Law, and the resulting equations are second order differential equations. These equations may be reduced to first order equations and hence, the problem is formulated in terms of a first order, nonlinear, ordinary, vector differential equation

$$\dot{x} = f(x,u,t) \tag{2.1}$$

where x is an n vector of state variables, f is an n vector of known functions, u is an m vector of control variables, and t is the independent variable time. The performance index, which is the function to be extremized, is a scalar

$$\phi = \phi(x_f,t_f) \tag{2.2}$$

and is a function of terminal state and time. The specified initial constraint relations are

$$\eta = \eta(x_0,t_0) = 0 \tag{2.3}$$

where η is a p vector, and the specified terminal constraint relations are

$$\psi = \psi(x_f,t_f) = 0 \tag{2.4}$$

where Ψ is a q vector.

The classical method of extremizing a function while satisfying specified terminal constraints is to adjoin the constraints and the constraining differential equations of motion to the functional with the Lagrange multipliers ν^T and λ^T, respectively. The functional to be extremized becomes

$$I = \phi(x_f,t_f) + \nu^T\Psi(x_f,t_f) \tag{2.5}$$

$$+ \int_{t_0}^{t_f} \lambda^T(t)[f(x,u,t) - \dot{x}]dt$$

where ϕ is the scalar performance index, ν is a q vector of constant Lagrange multipliers, Ψ is a q vector of specified terminal constraint relations, and λ is an n vector of time dependent Lagrange multipliers. Eq. (2.3) is usually easily solved for p of the initial conditions needed to integrate Eq. (2.1).

The functional I is simplified by introducing a quantity P where $P = \phi(x_f,t_f) + \nu^T\Psi(x_f,t_f)$ and the generalized Hamiltonian $H = \lambda^T(t)f(x,u,t)$. The functional I becomes

$$I = P(x_f,t_f) - \int_{t_0}^{t_f} (\lambda^T\dot{x} - H)dt \,. \tag{2.6}$$

The first term under the integral sign may be integrated by parts and the functional rewritten

$$I = P\Big|_{t_0}^{t_f} - \lambda^T x\Big|_{t_0}^{t_f} + \int_{t_0}^{t_f} (\dot{\lambda}^T x + H)dt \ . \qquad (2.7)$$

The functional is now expanded in a Taylor series about some nominal trajectory such that $dI = dI' + dI'' + \ldots$ where the term dI' designates the first variation, the second term dI'', the second variation and so forth. The first variation dI' is given by

$$dI' = dP\Big|_{t_0}^{t_f} - d(\lambda^T x)\Big|_{t_0}^{t_f} + d\int_{t_0}^{t_f} (\dot{\lambda}^T x + H)dt \qquad (2.8)$$

and taking the total differential of each term and using Leibnitz's Rule on the last term, the equation becomes

$$dI' = (P_x dx + P_\nu d\nu + P_t dt)\Big|_{t_0}^{t_f} - (d\lambda^T x + \lambda^T dx)\Big|_{t_0}^{t_f}$$
$$+ (\dot{\lambda}^T x + H)dt\Big|_{t_0}^{t_f} + \int_{t_0}^{t_f} [\delta\dot{\lambda}^T x + \dot{\lambda}^T \delta x + \delta\lambda^T f + \lambda^T (f_x \delta x + f_u \delta u)]dt \ . \qquad (2.9)$$

Integrating the first term under the integral sign by parts and noting that to first order $d\lambda_i^T = \delta\lambda_i^T + \dot{\lambda}_i^T dt_i$, where $i = 0$ or f, the Eq. (2.9) may be rewritten. After collecting the terms that must be evaluated at the initial and terminal times, and making the appropriate cancellations, the Eq. (2.9) becomes

$$dI' = [(P_x - \lambda^T)dx + P_\nu d\nu + (P_t + H)dt]\Big|_{t_0}^{t_f}$$
$$+ [\lambda^T dx - Hdt]\Big|_{t_0} + \int_{t_0}^{t_f} [\delta\lambda^T(f-\dot{x}) + (\dot{\lambda}^T+H_x)\delta x + H_u \delta u]dt \ . \tag{2.10}$$

The first necessary conditions for the functional I and hence for the performance index ϕ to be extremized is that the first variation dI' must vanish. The vanishing of the first variation implies that each term in Eq. (2.10) must vanish if the variations dx_f, $d\nu$, dt_f, dx_0, dt_0, $\delta\lambda$, δx and δu are independent variations. Therefore, the necessary conditions that must be satisfied at the initial boundary are as follows:

(1) $$\lambda^T dx\Big|_{t_0} = 0 \tag{2.11}$$

This condition implies that if the initial state is specified, i.e. $dx(t_0) = 0$, the equation is identically satisfied. If, however, the initial state is unspecified, the associated Lagrange multipliers must vanish at the initial time. This assumes that the initial state and time variations are independent of one another, and if they are Eq. (2.11) yields n initial conditions.

$$(2) \qquad -H dt\Big|_{t_0} = 0 \qquad (2.12)$$

This condition implies that if the initial time is specified, i.e. $dt_0 = 0$, the equation is identically satisfied. If, however, the initial time is unspecified and the initial state and time variations are independent of one another, the generalized Hamiltonian $\lambda^T f$ must vanish at the assumed initial time. This yields one initial condition.

The necessary conditions that must be satisfied at the terminal boundary are as follows:

$$(1) \qquad P_\nu d\nu\Big|^{t_f} = 0 \qquad (2.13)$$

This condition implies that $\Psi d\nu\Big|^{t_f} = 0$ since $\frac{\partial P}{\partial \nu} = \Psi$. The specified terminal constraints must be satisfied, and hence the $d\nu$ does not necessarily vanish. This yields q terminal conditions, $\Psi = 0$.

$$(2) \qquad (P_x - \lambda^T)dx\Big|^{t_f} = 0 \qquad (2.14)$$

This condition implies that if the terminal state is unspecified, the coefficient $(\phi_x + \nu^T \Psi_x - \lambda^T)\Big|^{t_f}$ must vanish. This transversality condition yields n terminal conditions.

(3) $$(P_t + H)dt\Big|^{t_f} = 0 \qquad (2.15)$$

This condition implies that if the terminal time is unspecified, the coefficient $(\phi_t + \nu^T \Psi_t + H)\Big|^{t_f}$ must vanish. This transversality condition yields one terminal condition.

The necessary conditions that must be satisfied at every point along the trajectory are as follows:

(1) $$\dot{x} - f(x,u,t) = 0 \qquad (2.16)$$

This is the original nonlinear differential equation of motion and consists of n equations.

(2) $$\dot{\lambda}^T + H_x(\lambda,x,u,t) = 0 \qquad (2.17)$$

This equation is the classical Euler-Lagrange equation and consists of n equations.

(3) $$H_u(\lambda,x,u,t) = 0 \qquad (2.18)$$

This equation is the classical optimality condition and consists of m equations. This equation may also be recognized as the weak form of the Pontryagin Maximum Principle.

The problem is now theoretically solvable since the Eqs. (2.11) through (2.15) yield 2n+q+2 initial and terminal

boundary conditions for the $2n$ first order differential equations, Eqs. (2.16) and (2.17), and the $q+2$ unknowns ν, t_0, and t_f. The m control variables may either be eliminated from Eqs. (2.16) and (2.17) by using the optimality condition Eq. (2.18), or Eq. (2.18) may be differentiated and treated as another differential equation. In this case

$$\frac{d}{dt}\left[H_u(\lambda,x,u,t)\right] = 0 \tag{2.19}$$

and expanding Eq. (2.19) leads to the expression

$$H_u\dot{\lambda} + H_{ux}\dot{x} + H_{uu}\dot{u} + H_{ut} = 0 \ . \tag{2.20}$$

By inverting the H_{uu} matrix, the time rate of change of the control vector becomes

$$\dot{u} = -H_{uu}^{-1}\left[H_{u\lambda}\dot{\lambda} + H_{ux}\dot{x} + H_{ut}\right] \ . \tag{2.21}$$

Using the differential equations of motion, Eq. (2.16) and the Euler-Lagrange equations, Eq. (2.17), Eq. (2.21) becomes

$$\dot{u} = -H_{uu}^{-1}\left[H_{ux}H_{\lambda}^{T} - H_{u\lambda}H_{x}^{T} + H_{ut}\right] \tag{2.22}$$

which may be simultaneously integrated with Eqs. (2.16) and (2.17).

However, for such an integration, an initial condition for the control must be known. The optimality condition

yields the control in terms of the state and Euler variables, and since these parameters must either be assumed or known initially anyway, the initial condition on the control may be determined easily.

The justification for the statement that $H_u = \dot{H}_u = 0$ (and for that matter $\ddot{H}_u = \dddot{H}_u = \ldots = 0$) is that the optimality condition $H_u = 0$ must be identically satisfied at every point along the optimal trajectory and at no point can there be a deviation from $H_u = 0$.

The previously stated first necessary conditions are the ones necessary for the functional I to assume a stationary value, however these conditions are not sufficient to insure that a minimum has been obtained. If the Legendre Condition is satisfied and if no conjugate points exist in the interval of the independent variable, the fourth necessary condition, and the one that is sufficient to insure a strong minimum, involves the Weierstrass E-Function. The E-Function is explained by Gelfand (27) and must be equal to or greater than zero for a minimum. An application of the Weierstrass E-Function is shown in Appendix A.1 for a vehicle moving in an inverse square gravitational force field under the influence of a thrust force.

2.2 Reduction of the Optimization Problem to a Two-Point Boundary Value Problem

The classical trajectory optimization problem may be reduced to a two-point boundary value problem and hence

several previously known methods become available for its solution. The first necessary conditions previously derived in Section 2.1 must be used, and frequent reference is made to that section. The conditions that must be satisfied at every point along the trajectory are Eqs. (2.16), (2.17), and (2.18), i.e. the differential equations of motion

$$\dot{x} = f(x,u,t) \tag{2.23}$$

where x is an n vector of state variables, the differential equation that is adjoint to the linearized differential equation of motion and called the Euler-Lagrange equation

$$\dot{\lambda} = -f_x^T\lambda = -H_x^T(\lambda,u,x,t) \tag{2.24}$$

where λ is an n vector of adjoint variables, and the classical optimality condition

$$H_u(\lambda,u,x,t) = 0 \tag{2.25}$$

where H is the generalized Hamiltonian and u is an m vector of control variables.

The m Eqs. (2.25) may be solved for the m unknown control variables in terms of the state and adjoint variables and time, and the control then eliminated from Eqs. (2.23) and (2.24).

In the general case, where the initial state and time variations are not independent of one another, Eqs. (2.11) and (2.12) must remain as one equation. Hence, the initial

conditions that must be satisfied are the initially specified constraint relations, Eq. (2.3)

$$\eta(x_0,t_0) = 0 \tag{2.26}$$

where η is a p vector, and the transversality condition

$$(\lambda^T dx - H dt)\Big|_{t_0} = 0 \ . \tag{2.27}$$

The state and time total variations dx_0 and dt_0 are not necessarily independent of one another, and in fact are related through Eq. (2.6). It is required that for all dx_0 and dt_0 that $d\eta(x_0,t_0) = 0$, and to a first order approximation this condition can be expressed as

$$\left[\frac{\partial \eta}{\partial x}\right]_0 dx_0 + \left[\frac{\partial \eta}{\partial t}\right]_0 dt_0 = 0 \tag{2.28}$$

Since $d\eta(x_0,t_0)$ is a p vector of conditions, it follows that p of the $n+1$ total variations dx_0 and dt_0 may be determined in terms of the remaining $n+1-p$ variations. These p total variations are eliminated from the variations in Eq. (2.27), leaving $n+1-p$ independent variations. The coefficients of these $n+1-p$ independent variations may be equated to zero to obtain $n+1-p$ additional relations at the initial time. Combining these $n+1-p$ relations with the p initially specified constraint relations in Eq. (2.26) will result in the desired $n+1$ initial conditions, $g(x_0,t_0) = 0$ and t_0 .

In most cases, the initial state and time are given, which would be the required $n+1$ conditions, and the transversality condition Eq. (2.27) is then identically satisfied.

The terminal conditions that must be satisfied are the terminally specified constraint relations, Eq. (2.13)

$$\Psi(x_f, t_f) = 0 \tag{2.29}$$

where Ψ is a q vector, and the transversality conditions, Eqs. (2.14) and (2.15),

$$(P_x - \lambda^T)dx\Big|^{t_f} = 0 \tag{2.30}$$

$$(P_t + H)dt\Big|^{t_f} = 0 \ . \tag{2.31}$$

Since the Lagrange multipliers ν were introduced, the total variations, dx_f and dt_f, in Eqs. (2.30) and (2.31) can be treated as independent variations, and the coefficients of these variations may be equated to zero. This procedure provides $n+1$ terminal conditions, n resulting from Eq. (2.30) and one from Eq. (2.31). There are, however, q remaining unknowns to be evaluated, i.e. the q Lagrange multipliers ν. The q terminally specified constraints given in Eq. (2.29) provide the additional conditions for this operation.

In summary, the terminal conditions become

$$h_i = \psi_i(x_f,t_f) \qquad \text{for} \qquad i = 1,q \qquad (2.32)$$

$$h_i = (\phi_x+\nu^T\psi_x-\lambda^T)_i \qquad \text{for} \qquad i = q+1,\ n+q \qquad (2.33)$$

and

$$h_i = (\phi_t+\nu^T\psi_t+H)_i \qquad \text{for} \qquad i = n+q+1\ . \qquad (2.34)$$

The $n+1$ initial conditions are combined with the $n+q+1$ terminal conditions to obtain the boundary conditions for the $2n^{th}$ order system of differential equations given by Eqs. (2.23) and (2.24), t_0, t_f, and the q values of ν.

If the terminal constraint relations are not very complicated, it may be easier to eliminate the Lagrange multipliers ν from the start. Hence, an alternative approach, which considers the functional

$$I = \phi + \int_{t_0}^{t_f} \lambda^T(f - \dot{x})dt\ ,$$

would yield transversality conditions

$$(\phi_x-\lambda^T)dx\Big|_{t_0}^{t_f} + (\phi_t+H)dt\Big|_{t_0}^{t_f} = 0 \qquad (2.35)$$

to be satisfied.

However, the total variations dx_f and dt_f are not independent, and are related in fact through the terminally specified constraint relation, Eq. (2.29). It is required that $d\psi(x_f,t_f) = 0$, and to a first order approximation this becomes

$$\left[\frac{\partial \Psi}{\partial x}\right]_f dx_f + \left[\frac{\partial \Psi}{\partial t}\right]_f dt_f = 0 \qquad (2.36)$$

where $d\Psi(x_f,t_f)$ is a q vector. Now q of the n+1 total variations dx_f and dt_f may be determined in terms of the remaining n+1-q variations. These q total variations are eliminated from the variations in Eq. (2.35), leaving only n+1-q independent variations. The coefficients of these n+1-q independent variations may be equated to zero thus obtaining n+1-q relations at the terminal time. Combining these n+1-q relations with the q terminally specified constraint relations Eq. (2.29), will lead to the desired n+1 terminal conditions, $h(x_f,t_f) = 0$. This procedure of eliminating the Lagrange multipliers ν , requires the determination of q less parameters in the iteration procedure for solving the two-point boundary value problem.

The complete solution of the two-point boundary value problem requires 2n+1 boundary conditions, assuming that the initial time is given, and these conditions may be derived in the manner described above. To reduce the number of parameters that require determination, it is assumed that the terminal constraint relations are included without the use of the Lagrange multipliers ν . Furthermore, it is assumed that the control variables are eliminated from Eqs. (2.23) and (2.24), by using the optimality condition, Eq. (2.25).

In summary, the problem is formulated in term of an ordinary, first order, nonlinear, vector differential equation

$$\dot{z} = F(z,t) \tag{2.37}$$

where z is a $2n$ vector composed of n state variables and n Euler-Lagrange variables and t is the independent variable time. More specifically,

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} H_{\lambda}^{T}(x,\lambda,t) \\ -H_{x}^{T}(x,\lambda,t) \end{bmatrix} = F(z,t) \ . \tag{2.38}$$

It is assumed that p initially specified constraint relations

$$\eta(z_0,t_0) = 0 \tag{2.39}$$

and a specified initial time t_0 are given. Since these conditions are given, only $n-p$ initial relations must be obtained from the transversality condition, Eq. (2.27) and hence a total of n conditions at the initial time are known. These n conditions are represented as

$$g(z_0,t_0) = 0 \tag{2.40}$$

Consider that q terminally specified constraint relations

$$\psi(z_f,t_f) = 0 \tag{2.41}$$

are given. This implies that $n+1-q$ terminal relations must be obtained from the transversality condition, Eq. (2.35), which when combined with Eq. (2.41) yields $n+1$ terminal constraint relations

$$h(z_f, t_f) = 0 \tag{2.42}$$

The $2n+1$ conditions needed for the two-point boundary value problem solution are specified, n conditions from Eq. (2.40) and $n+1$ conditions from Eq. (2.42).

An application of the reduction of an optimization problem to a two-point boundary value problem is shown in Appendix A.1.

CHAPTER 3

PERTURBATION METHODS

Several of the most promising and successful methods for solving the nonlinear two-point boundary value problem, associated with the optimization of spacecraft trajectories, are classified as Perturbation Methods. These methods are sometimes referred to as Second Variation or Extremal Field Methods.

The Perturbation Methods are divided into two groups, the Methods of Adjoint Functions and the Method of Perturbation Functions. The Method of Perturbation Functions require the use of functions obtained through a linear perturbation about some nominal path, while the Method of Adjoint Functions require the use of functions which are adjoint to the perturbation functions. The adjoint functions, along with the perturbation functions, are used to approximate the influence of initial variable variations on terminal variable variations.

The theoretical development of the Method of Adjoint Functions and the Method of Perturbation Functions may be shown to follow common lines and in this sense the formulations are parallel. For the special case discussed later, the two methods in fact become the same.

As discussed in Chapter 2, the optimization problem is formulated in terms of an ordinary, first order, nonlinear, vector differential equation

$$\dot{z} = F(z,t) \tag{3.1}$$

where $\dot{z}$ and $F(z,t)$ are partitioned as shown in Eq. (2.38).

The perturbation equations are derived by making a linear expansion of Eq. (3.1) about some nominal path. These equations are represented by

$$\delta\dot{z} = \left[\frac{\partial F}{\partial z}\right]\delta z = A\delta z \tag{3.2}$$

where δz is a $2n$ vector of state and Euler-Lagrange variable variations and the $2n \times 2n$ matrix of partial derivatives A is evaluated along the nominal path. The equations that govern the set of functions adjoint to the perturbation equations, Eq. (3.2) are

$$\dot{\Lambda} = -\left[\frac{\partial F}{\partial z}\right]^T \Lambda = -A^T\Lambda \tag{3.3}$$

where Λ is a $2n$ vector of adjoint variables. The motivation for the use of this equation becomes evident when Eq. (3.8) is developed.

In the general case, the nominal trajectory will not satisfy the $n+1$ terminal constraint relations on the first iteration because all the proper initial conditions are not known. To obtain a relation for the terminal constraint

dissatisfaction as a function of the total terminal variations, $dz(t_f)$ and dt_f, the Eq. (2.42) is perturbed about the nominal terminal conditions, to obtain

$$dh = \left[\frac{\partial h}{\partial z}\right]_f dz_f + \left[\frac{\partial h}{\partial t}\right]_f dt_f \tag{3.4}$$

where dh is an $n+1$ vector of the change of the dissatisfaction in the terminal constraint relations, $\left[\frac{\partial h}{\partial z}\right]_f$ is an $n+1 \times 2n$ matrix of partial derivatives, and $\left[\frac{\partial h}{\partial t}\right]_f$ is an $n+1$ vector of partial derivatives.

If allowance is made for the possibility of a state and/or Euler variable variation resulting from a terminal time variation, the following first order relation may be made

$$dz(t_f) = \delta z(t_f) + \dot{z}(t_f)dt_f \,. \tag{3.5}$$

When this relation is substituted into the perturbed terminal constraint relations, Eq. (3.4), and a rearrangement is made, the resulting equation becomes

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \delta z(t_f) + \dot{h}dt_f \tag{3.6}$$

where dh is an $n+1$ vector of terminal dissatisfaction change. This relation is an indication of how the terminal constraint dissatisfaction change is affected by variations in the terminal values of state and Euler variables and total variations in terminal time.

It may be noted here that if the terminal variation of $z(t_f)$ is determined as some linear function of the initial variation of $z(t_0)$, i.e. $\delta z(t_f) = [\pi]\delta z(t_0)$, where π is some $2n \times 2n$ matrix, the terminal dissatisfaction change becomes a function of the initial state and Euler variable variation $\delta z(t_0)$ and the terminal time variation dt_f. This substitution results in

$$dh = \left[\frac{\partial h}{\partial z}\right]_f [\pi]\delta z(t_0) + \dot{h}dt_f \quad . \tag{3.7}$$

An iteration procedure may now be designed to reduce the terminal dissatisfaction by proceeding in the following manner:

(1) Integrate the nonlinear differential equations, Eq. (3.1), forward from t_0 to some assumed terminal time t_f, using the n known initial conditions given by Eq. (2.40) and assuming n initial values for the remaining variables.

(2) When the assumed terminal time t_f is reached, the matrix $\left[\frac{\partial h}{\partial z}\right]_f$, the vector $\dot{h}$ and the terminal constraint dissatisfaction change dh may be determined.

(3) The terminal dissatisfaction may be reduced on the next iteration by requesting that some percentage

of the present dissatisfaction be corrected, i.e. $dh = -ch$, where $0 \leqslant c \leqslant 1$.

(4) Determination of $[\Pi]\delta z(t_0)$ must be made in some manner and will be discussed in the next sections.

(5) The linear algebraic equations, Eq. (3.7), are solved for the corrections $\delta z(t_0)$ and dt_f, and these values are applied to the initially assumed values of $z(t_0)$ and t_f .

(6) The procedure is repeated until the corrections being applied are less than some preselected value.

The only remaining theoretical problem is to determine $[\Pi]\delta z(t_0)$, and the manner in which this is done determines whether the technique is classified as a Method of Adjoint Functions or Perturbation Functions. Techniques for determining $[\Pi]\delta z(t_0)$ are discussed in the following sections.

3.1 Methods of Adjoint Functions

There are several methods of determining the terminal state and Euler variable variations as a function of the initial variations, i.e. $\delta z(t_f) = [\Pi]\delta z(t_0)$. A relation that contains these two variations may be derived by premultiplying the perturbation equation, Eq. (3.2), by the transpose of the adjoint vector Λ , and postmultiplying the transpose of the adjoint equations, Eq. (3.3), by δz and adding the resulting equations to obtain

$$\frac{d}{dt}(\Lambda^T \delta z) = 0 \ . \tag{3.8}$$

This equation may be integrated from t_0 to t_f to obtain

$$\Lambda^T(t_f)\delta z(t_f) = \Lambda^T(t_0)\delta z(t_0) \tag{3.9}$$

where the boundary conditions on the adjoint variables are completely arbitrary and may be selected such that the desired relationship between $\delta z(t_f)$ and $\delta z(t_0)$ is obtained. There are several approaches that may be taken.

The first approach and a most natural one is to integrate the adjoint equations, Eq. (3.3), backwards from t_f to t_0, $2n$ times with the starting conditions

$$ {}_1\Lambda_1^T(t_f) \ , \ {}_1\Lambda_2^T(t_f) \ \ldots \ {}_1\Lambda_{2n}^T(t_f) \quad \text{or} \quad {}_1\theta(t_f,t_f)$$

where

$$ {}_1\theta(t_f,t_f) = \begin{bmatrix} {}_1\Lambda_1^T(t_f) \\ {}_1\Lambda_2^T(t_f) \\ \vdots \\ {}_1\Lambda_{2n}^T(t_f) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = I \ . \tag{3.10}$$

The presubscript refers to the first approach. When this integration is completed, Eq. (3.9) may be written

$$\delta z(t_f) = {}_1\Theta(t_f,t_0)\,\delta z(t_0) \qquad (3.11)$$

Substituting this equation into the perturbed terminal constraint relation, Eq. (3.6), yields the desired relation

$$dh = \left[\frac{\partial h}{\partial z}\right]_f {}_1\Theta(t_f,t_0)\,\delta z(t_0) + \dot{h}\,dt_f \qquad (3.12)$$

where

dh is an $n+1$ vector representing the change in the terminal dissatisfaction.

$\left[\frac{\partial h}{\partial z}\right]_f$ is an $n+1 \times 2n$ matrix evaluated at the nominal terminal time, t_f.

${}_1\Theta(t_f,t_0)$ is an $2n \times 2n$ matrix resulting from the $2n$ backward integrations of the adjoint equations.

$\delta z(t_0)$ is a $2n$ vector of initial variable variations that along with dt_f produce the terminal dissatisfaction change.

$\dot{h}$ is an $n+1$ vector which represents the time rate of change of the terminal dissatisfaction, evaluated at the nominal terminal time, t_f.

dt_f is a scalar variation of the nominal terminal time.

It must be noted that all of the perturbations $\delta z(t_0)$ are not independent, but in fact are related through the initial constraint relations Eq. (2.40). Assuming that the initial time is specified, the required first order expansion of Eq. (2.40) becomes

$$dg = \left[\frac{\partial g}{\partial z}\right]_0 \delta z(t_0) = 0 \tag{3.13}$$

This equation may be solved for n of the $\delta z(t_0)$ in terms of the remaining n elements of $\delta z(t_0)$, and these variations are eliminated from Eq. (3.12). This leaves the $n+1$ Eqs. (3.12) with the n independent $\delta z'(t_0)$ and terminal time variation dt_f as the $n+1$ unknowns. The prime indicates that the vector has been reduced in dimension so that only independent variations remain.

This approach is fundamental and very inefficient, because more information is generated than needed. The computational difficulties associated with the backwards integration of the adjoint equations may be eliminated by considering a second approach.

This approach requires the forward integration of the adjoint equations $2n$ times from t_0 to t_f with the starting conditions ${}_2\Lambda_1^T(t_0)$, ${}_2\Lambda_2^T(t_0)$ ... ${}_2\Lambda_{2n}^T(t_0)$ or ${}_2\Theta(t_0,t_0)$ where

$$ {}_2\Theta(t_0,t_0) = \begin{bmatrix} {}_2\Lambda_1^T(t_0) \\ {}_2\Lambda_2^T(t_0) \\ \vdots \\ {}_2\Lambda_{2n}^T(t_0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = I \quad . \qquad (3.14) $$

The presubscript refers to the second approach. When this integration is completed (and it may be performed simultaneously with the integration of Eq. (3.1), Eq. (3.9) becomes

$$ {}_2\Theta(t_0,t_f)\delta z(t_f) = \delta z(t_0) $$

and solving for $\delta z(t_f)$ yields

$$ \delta z(t_f) = [{}_2\Theta(t_0,t_f)]^{-1}\delta z(t_0) \quad . \qquad (3.15) $$

Substituting this equation into the perturbed terminal constraint relation, Eq. (3.6), yields the desired relation

$$ dh = \left[\frac{\partial h}{\partial z}\right]_f [{}_2\Theta(t_0,t_f)]^{-1}\delta z(t_0) + \dot{h}dt_f \qquad (3.16) $$

where the terms have the same physical significance as in the first approach.

The obvious disadvantage with this second approach is that even though the backward integration has been eliminated, the same number of equations must be integrated and a $2n \times 2n$ matrix must be inverted at the terminal time. It would

certainly be desirable if an approach could be formulated such that the above matrix inversion is unnecessary and a more efficient integration is made.

The third approach requires the examination of Eq. (3.12) which results from the first approach. Since the initial conditions on the linear adjoint equation, Eq. (3.3), are arbitrary and may be selected for convenience, an equation identical to Eq. (3.12) may be derived by integrating the adjoint equations only $n+1$ times with the starting conditions

$$\Theta(t_f,t_f) = \left[\frac{\partial h}{\partial z}\right]_f \tag{3.17}$$

where $\left[\frac{\partial h}{\partial z}\right]_f$ is an $n+1 \times 2n$ matrix evaluated at the nominal terminal time. In other words, since the linear adjoint equation is integrated with starting conditions ${}_1\Theta(t_f,t_f) = I$ in the first approach and results in ${}_1\Theta(t_f,t_0)$, if the starting condition were ${}_1\Theta(t_f,t_f) = \left[\frac{\partial h}{\partial z}\right]_f I$, the result would be $\left[\frac{\partial h}{\partial z}\right]_f {}_1\Theta(t_f,t_0)$. Hence, Eq. (3.12) has been derived with $n-1$ fewer integrations of an equivalent set of equations.

For this last approach the desired equation may be written

$$dh = \Theta(t_f,t_0)\delta z(t_0) + \dot{h}dt_f \tag{3.18}$$

where the terms have the same physical significance as the previous two approaches, but $\Theta(t_f,t_0)$ is an n+1 x 2n matrix resulting from the simultaneous backward integration of the adjoint equations. Again the dependent initial state and/or Euler variable variations must be eliminated, and this leaves n initial variable variations and one terminal time variation to be determined from the n+1 equations, Eq. (3.18).

The explanation for the third approach gives the justification for the scheme used by Jazwinski (12) where an extension is made of Jurovics and McIntyre's (7) presentation. One additional time conserving feature, which may be used, is the scaling of the Lagrange multipliers. This advantage results because the Euler-Lagrange equations are linear and homogeneous. The implementation of this idea is discussed in Section 7.3 and essentially involves the trading of one terminal condition for an initial condition. The decrease in the dimension of the terminal constraint vector by one, also decreases the number of adjoint integrations by one, and hence results in less computation time.

One additional remark is in order for cases where the specified terminal constraints are rather complex and the Lagrange multiplier ν is introduced. For this case, the terminal constraint vector becomes

$$h = h(z_f,t_f,\nu) = 0 \tag{3.19}$$

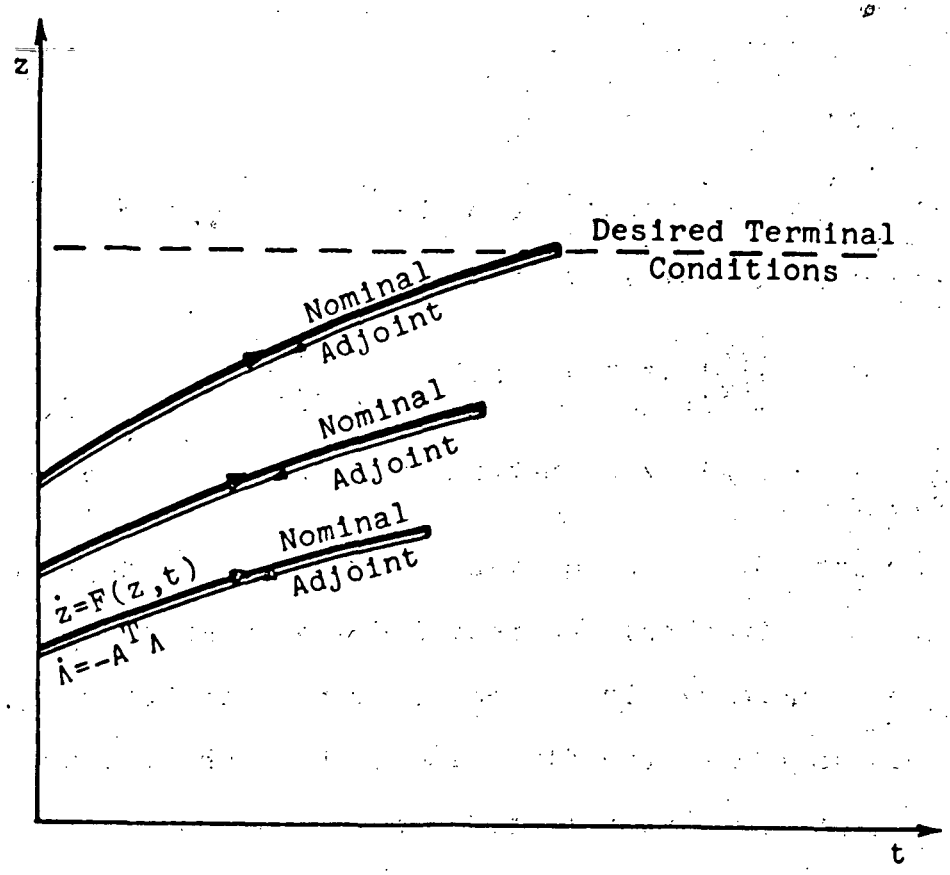
where h is an $n+1+q$ vector, and the perturbed terminal constraint relation, Eq. (3.4), becomes

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \delta z(t_f) + \dot{h} dt_f + \left[\frac{\partial h}{\partial \nu}\right]_f d\nu \qquad (3.20)$$

where $\left[\frac{\partial h}{\partial \nu}\right]_f$ is an $n+1 \times q$ matrix evaluated at the nominal terminal time and $d\nu$ is a q vector of total Lagrange multiplier variations. It should be recalled that when the ν vector is used, there exists $n+1+q$ terminal constraint relations and this increases the dimension of the dh vector by q. This is just the number of additional equations needed to solve for the additional unknown variations $d\nu$. These variations are applied to the assumed values of ν.

A similar technique is used by Breakwell, Speyer, and Bryson (9). It is shown in this reference that after the forward integration of Eq. (3.1) has been made, q of the n equations represented by Eq. (2.33) may be used to determine the q values of ν. Then these q values of ν are used to evaluate the terminal dissatisfaction represented by the remaining $n-q$ equations of Eq. (2.33). This procedure simply reduces the dimension of h to $n+1$, and hence only $n+1$ backward integrations of Eq. (3.3) are needed.

The computational procedure may be followed by referring to an illustration of the Method of Adjoint Functions (MAF):



(1) Integrate the $2n$ nonlinear differential equations of motion and the Euler-Lagrange equations, Eq. (3.1), forward from t_0 to t_f with starting conditions satisfying Eq. (2.40) and n assumed values for the unknown parameters.

(2) Evaluate at the nominal terminal time, t_f, the quantities $\dot{h}$, h, and the starting conditions for the backwards integration of the adjoint equations, $\left[\frac{\partial h}{\partial z}\right]_f$.

(3) Integrate the $2n$ adjoint equations, Eq. (3.3), backwards $n+1$ times from t_f to t_0 with starting conditions, $\left[\frac{\partial h}{\partial z}\right]_f$ and use the value of the variables

stored during the forward integration to form the coefficients of the adjoint variables.

(4) Solve the n+1 linear algebraic equations, Eqs. (3.18), for a linear approximation of the corrections that must be applied to the assumed initial values and the terminal time.

(5) Apply these corrections and repeat the process until the corrections become smaller than some pre-selected value.

3.2 Methods of Perturbation Functions

Of the several methods available for determining the terminal variations in the state and Euler variables as a function of the initial variations, i.e. $\delta z(t_f) = [\Pi]\delta z(t_0)$, the most natural one involves the direct use of the perturbation equations, Eq. (3.2)

$$\dot{\delta z} = A\delta z \ . \tag{3.21}$$

As a first approach, integrate these perturbation equations forward from t_0 to t_f, 2n times with the starting conditions

$$ {}_1\delta z_1(t_0),\ {}_1\delta z_2(t_0)\ \ldots,\ {}_1\delta z_{2n}(t_0) \quad \text{or} \quad {}_1\Phi(t_0,t_0) $$

where $_1\Phi(t_0,t_0) =$

$$[{}_1\delta z_1(t_0), {}_1\delta z_2(t_0), \ldots, {}_1\delta z_{2n}(t_0)] = \begin{bmatrix} 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ 0 & 0 & & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & & 1 \end{bmatrix} = I. \tag{3.22}$$

The presubscript refers to the first approach. This integration may be made simultaneously with the forward integration of the differential equations, Eq. (3.1), and hence less computer storage is required. When this integration is completed, the resulting equations evaluated at the terminal time may be represented by

$$\delta z(t_f) = {}_1\Phi(t_0,t_f)\delta z(t_0) \tag{3.23}$$

where $_1\Phi(t_0,t_f)$ is a 2n x 2n matrix of partial derivatives evaluated on the nominal trajectory. This equation may be substituted into the perturbed terminal constraint relation, Eq. (3.6), and the desired result becomes

$$dh = \left[\frac{\partial h}{\partial z}\right]_f {}_1\Phi(t_0,t_f)\delta z(t_0) + \dot{h}dt_f \tag{3.24}$$

where the symbols have been explained previously. These n+1 equations contain 2n initial state and Euler variable variations and one terminal time variation. However, the dependent variations may be eliminated as explained for the adjoint methods and only the n+1 independent variations must be determined.

This first approach, using the perturbation equations represents a very special case, because it can be shown to be the exact equivalent to the first approach using the adjoint equations. This can be shown by substituting into Eq. (3.9) the starting conditions

$$\delta z(t_0) = \Phi(t_0,t_0) = I$$
$$\Lambda^T(t_f) = \Theta(t_f,t_f) = I \ . \tag{3.25}$$

This substitution yields

$$\Phi(t_0,t_f) = \Theta(t_f,t_0) \tag{3.26}$$

and under these circumstances the algebraic equations for the adjoint method, Eq. (3.12), and the perturbation method, Eq. (3.24), become identical.

A second approach is suggested after examination of Eq. (3.24). Since the initial conditions on the linear perturbation equations, Eq. (3.21), are arbitrary and may be selected for convenience, an equation identical to Eq. (3.24) may be derived by integrating the perturbation equations only n+1 times with the starting conditions

$$_2\Phi(t_0,t_0) = \left[\frac{\partial h}{\partial z}\right]_f \tag{3.27}$$

where $\left[\frac{\partial h}{\partial z}\right]_f$ is an n+1 x 2n matrix evaluated at the nominal

terminal time. The resulting linear algebraic equation to be solved becomes

$$dh = {}_2\Phi(t_0,t_f)\delta z(t_0) + \dot{h}dt_f \qquad (3.28)$$

where ${}_2\Phi(t_0,t_f)$ is generated by only $n+1$ integrations of the perturbation equations.

This approach loses some appeal, however, when implementation begins because the starting condition, Eq. (3.27), cannot be evaluated until a nominal trajectory is integrated. Since the perturbation equations cannot be integrated simultaneously with the differential equations, the nominal path must be stored and no particular advantage over the adjoint method is realized.

A third approach, which proves to be the most efficient, may be formulated by observing the manner in which the ${}_1\Theta(t_f,t_0)$ and ${}_1\Phi(t_0,t_f)$ matrices are generated and used. For each of the n independent initial variations required a corresponding column of the ${}_1\Theta(t_f,t_0)$ or ${}_1\Phi(t_0,t_f)$ matrix is needed. Since the ${}_1\Theta(t_f,t_0)$ matrix is generated by rows, to determine any one column requires all $2n$ integrations of the adjoint equations. This, however, is not true for the perturbation methods, because the ${}_1\Phi(t_0,t_f)$ matrix is generated by columns. The elements of any n columns can be determined by simply integrating the perturbation equation n times, the starting vector having the element that corresponds to the

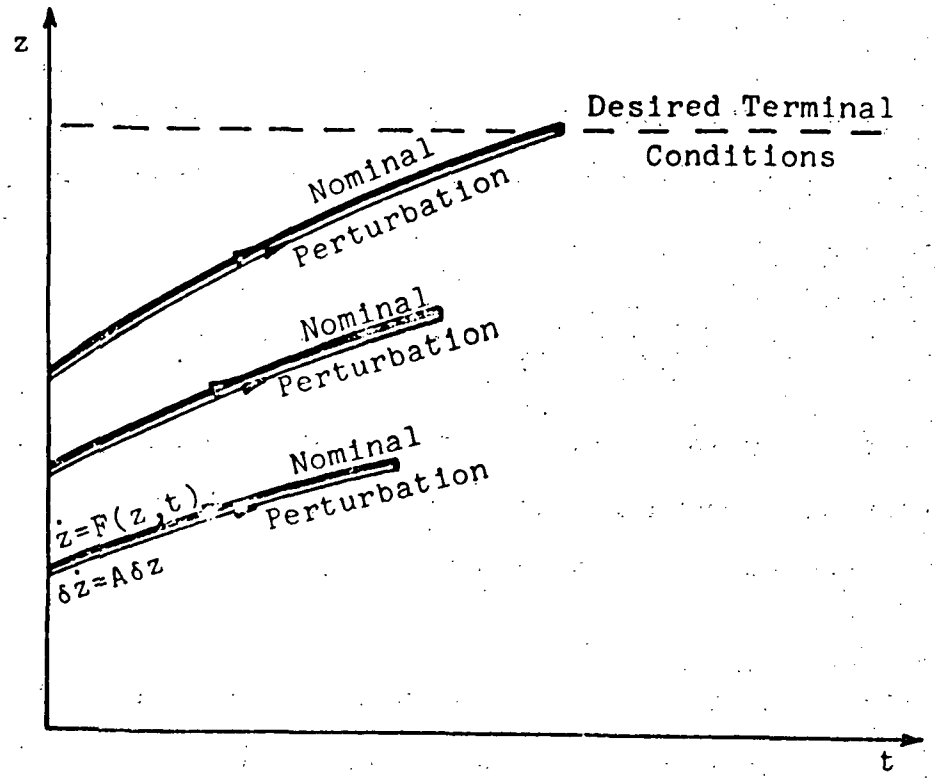
desired initial unknown variation set equal to unity and all others zero. With this modification, the linear algebraic equation becomes

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \Phi(t_0,t_f)\delta z'(t_0) + \dot{h}dt_f \qquad (3.29)$$

where $\Phi(t_0,t_f)$ is a $2n \times n$ matrix generated by integrating the perturbation equation only n times and $\delta z'(t_0)$ becomes an n vector representing the desired independent initial variations.

The essential feature of the perturbation method is that only n integrations are needed, and hence one less integration of a set of equations equivalent to the adjoint equations. The third approach to the adjoint method and the above perturbation method require the solution of exactly the same linear system, but the required elements of the $\Phi(t_0,t_f)$ matrix are simply derived in a more efficient manner. The additional advantage of using the perturbation method is that the nominal trajectory does not require computer storage.

The computational procedure may be followed by referring to an illustration of the Method of Perturbation Functions (MPF):



(1) Integrate the $2n$ nonlinear differential equations of motion and the Euler-Lagrange equations, Eq. (3.1), forward from t_0 to t_f with starting conditions consisting of the n known initial conditions satisfying Eq. (2.40) and n assumed values for the unknown parameters.

(2) Simultaneously with the above integration, integrate the $2n$ perturbation equations, Eq. (3.21), with starting conditions described above and coefficients formed from the variables that describe the nominal trajectory.

(3) Solve the n+1 linear algebraic equations, Eq. (3.29), for a linear approximation of the corrections that must be applied to the assumed initial values and the terminal time.

(4) Apply these corrections and repeat the process until the corrections become smaller than some preselected value.

3.3 Iteration Philosophy for the Perturbation Methods

The iteration schemes for the Perturbation Methods simply consist of a procedure for iteratively determining the initial values of the Lagrange multipliers so as to decrease the terminal constraint dissatisfaction on the following iteration. The control is eliminated from the differential equations, Eqs. (2.23) and (2.24), by using the optimality conditions, Eq. (2.25), and the nonlinear differential equations are integrated during each iteration. Since the optimality condition is always satisfied, every iteration produces an optimal trajectory, but to an undesired terminal condition. The only remaining complication is to satisfy the desired terminal constraints, Eq. (2.42).

Normally, the requested change in the terminal dissatisfaction is equated to the negative of the terminal dissatisfaction resulting from the previous iteration. This requested correction is then used in the linear algebraic equations, Eqs. (3.18) or (3.29), to make a multiple linear interpolation for

the variations of the initially assumed values of the state and/or Euler variables. When these corrections are applied and a new nominal trajectory integrated, the terminal constraint dissatisfaction is usually reduced.

The difficulty with this type of indirect optimization procedure is that when the terminal dissatisfaction is large, the linear approximations are not very representative of the nonlinear system, and the possibility for divergence is increased. The linearization is made about the current nominal trajectory, and whether or not this trajectory is close to satisfying the terminal constraints on any given iteration is immaterial. The essential factor is that the trajectory resulting in the next iteration be sufficiently near the previous one so that the linearization assumptions are not stretched beyond the limits of validity.

One natural approach, the motive for which resulted from a suggestion made by Breakwell, Speyer, and Bryson (9), is to request the correction of only a percentage of the terminal dissatisfaction resulting from the previous iteration. For instance, the algebraic equation that contains the corrections for the Method of Perturbation Functions is

$$dh = \left[\frac{\partial h}{\partial z}\right]_f \Phi(t_0,t_f)\delta z(t_0) + \dot{h}dt_f \qquad (3.30)$$

and for a percentage correction let

$$dh = -ch \qquad (3.31)$$

where c is the desired percentage to be corrected. The iteration factor c may have values in the range $0 \leqslant c \leqslant 1$. A correction for the Method of Adjoint Functions is applied in the same manner.

It is also reasonable to expect that as the optimal trajectory is approached, successive trajectories will be sufficiently near one another. Hence, the linear representation becomes accurate enough to request the complete correction of the terminal dissatisfaction. Also, as successive trajectory iterations begin to converge, successive adjoint and perturbation solutions begin to converge, and hence integration of these equations for every iteration may be unnecessary.

A summary and extension of the conjectures stated above, which result in some of the desired characteristics of an iteration scheme, are that:

(1) An iteration factor may be specified initially and changed during subsequent iterations by specifying an iteration rate factor. As the iterations proceed, the iteration rate factor is used to control the percentage of the terminal dissatisfaction corrected on any given iteration.

(2) There may exist an initial value of the iteration factor that minimizes the convergence time or maximizes the chance for convergence.

(3) It may be unnecessary to update the $\Phi(t_0,t_f)$ and $\Theta(t_f,t_0)$ matrices on every iteration.

(4) A correction of more than 100 percent may be reasonable and desirable.

These conjectures are investigated by using the following different iteration schemes:

Iteration Scheme 1 - This scheme for both the Methods of Adjoint and Perturbation Functions requires the arbitrary selection of an initial value of the iteration factor and the iteration rate factor. An iteration is made and the corresponding iteration factor is applied to obtain corrections for the next iteration. If the norm of the terminal dissatisfaction decreases on the next iteration, the iteration factor is increased by the value of the iteration rate factor. This process is repeated, never allowing the iteration factor to be zero or greater than unity, until the corrections for each assumed value is less than some preselected value.

A detailed procedure of Iteration Scheme 1 follows:

(1) Starting values of the iteration factor and the iteration rate factor are selected.

(2) Integrate the nonlinear differential equations of motion forward, noting the norm of the terminal dissatisfaction. If the Method of Adjoint Functions is

being used, integrate the adjoint equations backwards. If the Method of Perturbation Functions is being used, the perturbation equations may be integrated forward simultaneously with the differential equations of motion.

(3) Solve the algebraic equations, using the specified value of the iteration factor, to determine the corrections required for the initially assumed values.

(4) If all corrections are less than some preselected value, terminate the iteration. If any one correction is greater than the preselected value continue the process as follows.

(5) Apply the corrections to the assumed initial conditions, integrate the differential equations again, and determine the terminal dissatisfaction. If the norm of the terminal dissatisfaction is less than the norm that results on the previous iteration, increase the iteration factor by the value of the iteration rate factor and continue to iterate. Never allow the iteration factor to be greater than unity.

(6) If the norm is greater than the previous norm, decrease the iteration factor by the value of the iteration rate factor and continue to iterate. Never

allow the iteration factor to be less than the value of the iteration rate factor.

Iteration Scheme 2 - During the initial efforts to solve a problem with either the Method of Adjoint Functions or the Method of Perturbation Functions, a low initial value for the iteration factor is usually assumed. This requests a small change from a solution which is probably far from optimal, and thus reduces the possibility for divergence. However, this could be an unreasonably low estimate and if the iteration factor is systematically increased, as in Iteration Scheme 1, a great number of iterations would be required before a full correction would be requested. This scheme reduces the convergence time by avoiding the integration of the perturbation or adjoint equations on certain iterations. The criterion used to establish when a perturbation or adjoint equation integration is made is that either a divergence of the terminal constraint norm occurs or the integration is forced after a specified number of corrections have been made. The iteration factor is still increased each time a norm convergence occurs and the trajectory that produces this convergence is called a nominal. When the terminal norm diverges the iteration factor is decreased and the last convergent trajectory is used as a nominal.

A detailed procedure of Iteration Scheme 2 follows:

(1) Starting values of the iteration factor and the iteration rate factor are selected.

(2) Integrate the nonlinear differential equations of motion forward, noting the norm of the terminal dissatisfaction. If the Method of Adjoint Functions is being used, integrate the adjoint equations backwards. If the Method of Perturbation Functions is being used, the perturbation equations may be integrated forward simultaneously with the differential equations of motion.

(3) Solve the algebraic equations, using the specified value of the iteration factor, to determine the corrections required for the initially assumed values.

(4) If all corrections are less than some preselected value, terminate the iteration. If any one correction is greater than the preselected value continue the process as follows.

(5) Apply the corrections to the assumed initial conditions, integrate the differential equations again, and determine the terminal dissatisfaction. If the norm of the terminal dissatisfaction is less than the norm that results on the previous iteration, increase

the iteration factor by the value of the iteration rate factor. If the Method of Adjoint Functions is being used, avoid the adjoint integration on the present iteration. If the Method of Perturbation Functions is being used, avoid the perturbation integration on the next iteration.

(6) If the norm is greater than the previous norm, or if a specified number of iterations have been made, decrease the iteration factor by the value of the iteration rate factor. If the Method of Adjoint Functions is being used, the adjoint equations are integrated backwards where the coefficients are obtained from the last convergent forward trajectory. If the Method of Perturbation Functions is being used, the perturbation equations are integrated on the next iteration.

CHAPTER 4

QUASILINEARIZATION METHODS

The previously discussed Methods of Adjoint and Perturbation Functions involve the integration of a set of nonlinear differential equations. The coefficients for the linear adjoint or perturbation differential equations are formed with the variables generated by the nonlinear equations. A somewhat different approach can be formulated by linearizing the differential equations, and then using the adjoint and perturbation functions in the same general manner as before. The coefficients used to generate a new nominal trajectory are formed from the solution that corresponds to the previous nominal trajectory. This, essentially, is the quasilinearization concept.

The theoretical development of the Quasilinearization Methods may be shown to follow common lines, and in this sense the formulations are parallel. The approaches involve the solution of a set of linear differential equations, the solution of which converges, under appropriate conditions, to the solution of the desired nonlinear problem. Since the equations are linear, the terminal constraints can be satisfied on every iteration, if desired. However, the classical optimality condition is not satisfied until convergence has occurred, and even though the end points of the trajectory are satisfied, some care must be taken to insure that the trajectory shape

between these end points is correct. One other characteristic of the quasilinearization techniques is that an initially assumed solution is required. If a reasonable estimate of the solution cannot be made, a starting solution, derived from the integration of the nonlinear differential equations, may be good enough to result in convergence. This requires that only the initial values of the unknown variables be assumed, rather than the complete solution.

4.1 Methods of Generalized Newton-Raphson

The complete solution of the two-point boundary value problem by using the Method of Generalized Newton-Raphson may be obtained in a manner similar to the Method of Perturbation Functions discussed in Section 3.2. The exception to this similarity is that the differential equations, Eq. (3.1), are linearized about the previous nominal.

The problem is formulated in terms of an ordinary first order, nonlinear, vector, differential equation

$$\dot{z} = F(z,t) \tag{4.1}$$

where z is a $2n$ vector composed of n state variables and n Euler-Lagrange variables and t is the independent variable time. This nonlinear equation may be expanded about the previous nominal trajectory, say the n^{th} trajectory, and by ignoring the nonlinear terms yields

$$\dot{z}_{n+1} = \dot{z}_n + A(z_n,t)(z_{n+1} - z_n) \qquad (4.2)$$

where $A(z_n,t)$ is the partial derivative matrix $\left[\frac{\partial F}{\partial z}\right]_n$. This matrix is evaluated on the previous nominal trajectory and is similar to the $A(z,t)$ matrix discussed in the development of the Perturbation Methods. This equation, Eq. (4.2), can be expressed as

$$\dot{z} = Az + B \qquad (4.3)$$

where A is described above and $B = \dot{z}_n - Az_n$. Note that A and B are known from the previous nominal trajectory.

The first approach to the Method of Generalized Newton-Raphson is similar to the method outlined by McGill and Kenneth (13), and this provides a starting point for further development. Suppose that p of the initial values of z are specified, i.e. $z_i(t_0) = z_{i0}$, $i = 1, p$. This implies that $2n-p$ initial values of z must be assumed along with an assumed value of initial time t_0. The homogeneous part of Eq. (4.3) may be expressed as

$$\dot{y} = Ay \qquad (4.4)$$

and hence it is similar to the perturbation equations, Eq. (3.21). Eq. (4.4) may be integrated forward from t_0 to t_f $2n-p$ times with each successive starting vector consisting of

all zero elements except for the element that corresponds to one of the unknown initial conditions. This element is set equal to unity. This procedure leads to a $2n \times 2n-p$ matrix of solutions $Y(t_0,t)$. The forward integration amounts to making a unit perturbation in each one of the unknown initial conditions.

The nonhomogeneous solution to Eq. (4.3) may be obtained as a solution to

$$\dot{w} = Aw + B \qquad (4.5)$$

which generates a particular solution when integrated from t_0 to t_f with the p known initial conditions and $n-p$ assumed initial conditions. Now, the general solution of the linear system of Eqs. (4.3) becomes

$$z(t) = Y(t_0,t)C + w(t) \qquad (4.6)$$

where z is a $2n$ vector of state and Euler variables, Y is a $2n \times 2n-p$ matrix of homogeneous solutions, C is a $2n-p$ vector of constants and w is a $2n$ vector of nonhomogeneous solutions.

Since $2n+1-p$ conditions on the terminal value of z must be specified for a variable final time problem, any $2n-p$ of these conditions may be selected and the appropriate $2n-p$ members of Eq. (4.6) may be evaluated at the assumed terminal time. Then these equations are solved for the $2n-p$ constant corrections C. These corrections are used to update the

assumed initial conditions for the next iteration. For the purpose of saving computer storage the nominal trajectory is not formed by the linear combination of Eq. (4.6), but by integrating Eq. (4.3) with the updated initial conditions. This requires only the storage of the final values of the homogeneous and nonhomogeneous solutions.

This procedure is continued until a metric (that represents the maximum distance, over the complete independent variable range, between successive nominal trajectories) becomes less than some preselected value. This metric is given by

$$\rho = \sum_{i=1}^{N} \max_{t} \left| z_{n+1}^{i} - z_{n}^{i} \right| . \qquad (4.7)$$

Since this metric represents the maximum distance between successive nominal trajectories, its value decreases as the optimal trajectory shape is converged upon. When this metric has been reduced to an acceptable value, convergence has occurred for the specified value of terminal time. The one remaining unused terminal condition is used in a conventional scalar application of the Newton-Raphson iteration technique to produce a more accurate determination of terminal time. This finite difference equation is

$$t_{f_{k+1}} = t_{f_k} + \left[\frac{z_f - z\left(t_{f_k}\right)}{z\left(t_{f_k}\right) - z\left(t_{f_{k-1}}\right)} \right] \left(t_{f_k} - t_{f_{k-1}} \right) \qquad (4.8)$$

where the subscript k refers to the k^{th} time iteration and z_f is the desired terminal value of the variable selected. This new terminal time is used and trajectory iterations are made until the metric ρ is reduced once again. When the time iterations result in time changes smaller than some pre-selected value, the desired solution has been determined and the procedure is terminated.

One of the principal differences of the Method of Generalized Newton-Raphson as opposed to the Perturbation Methods is that an initial solution of the state and Euler variables is required. Also the method by which the terminal time is determined is very time consuming, especially when a large error is made in the assumed terminal time. A major objection is that the initial and terminal conditions must simply be values of the variables involved, rather than general functions of these variables. The above stated difference can be avoided, in some cases, by simply using the solution generated by integrating the nonlinear equations, Eq. (4.1), and this approach requires only starting values of the variables, p of which are known. The above stated objection has been partially removed by Long (16).

The method proposed by Long, designated here by the Modified Method of Generalized Newton-Raphson, involves a change of the independent variable

$$t = as \tag{4.9}$$

where a is a constant and s is a new independent variable having values $0 \leq s \leq 1$. The differential equations, Eq. (4.1), now become

$$z' = \frac{dz}{ds} = aF(z,as) . \qquad (4.10)$$

The constant a is considered a new state variable and an additional differential equation

$$a' = 0 \qquad (4.11)$$

may be added, but this is clearly not necessary since the solution to this equation is trivial. The value of a is initially assumed and then corrected on each iteration just like any other initially unknown state variable. The value a represents the terminal time as can be seen by evaluating Eq. (4.9) at the terminal value of the independent variable.

The determination of the terminal time now becomes an integral part of the iterative scheme, and its separate consideration, as required by the first approach, is not required. However, this does not save as much time as one might think, since a term that corresponds to the new state variable a must be added to each differential equation. Also another integration of the $2n$ homogeneous equations must be made since the value of a must be iteratively determined. The other objections discussed for the first approach are not

eliminated. The effectiveness of Long's proposal is evaluated and discussed further in a later chapter.

4.2 Modified Quasilinearization Method

The method proposed in the present study, called the Modified Quasilinearization Method, uses the quasilinearization concept but removes the restrictions on the Methods of Generalized Newton-Raphson discussed in Section 4.1. The manner in which the terminal time is determined proves superior to the modification proposed by Long.

The Eq. (4.6), derived for the Method of Generalized Newton-Raphson, can be rewritten and evaluated at the terminal time

$$Y(t_0,t_f)C = z(t_f) - w(t_f) \quad . \tag{4.12}$$

The right hand side of this equation is the difference between the desired terminal value of z and the linear calculation of the terminal value of w. This difference is interpreted as the variation of $z(t_f)$, and is expressed as $\delta z(t_f)$. Now, if both sides of Eq. (4.12) are premultiplied by $\left[\frac{\partial h}{\partial z}\right]_f$, the resulting expression becomes

$$\left[\frac{\partial h}{\partial z}\right]_f Y(t_0,t_f)C = \left[\frac{\partial h}{\partial z}\right]_f \delta z(t_f) \tag{4.13}$$

where $\left[\frac{\partial h}{\partial z}\right]_f$ is a $2n+1-p \times 2n$ matrix describing the partial change of a general set of terminal boundary conditions,

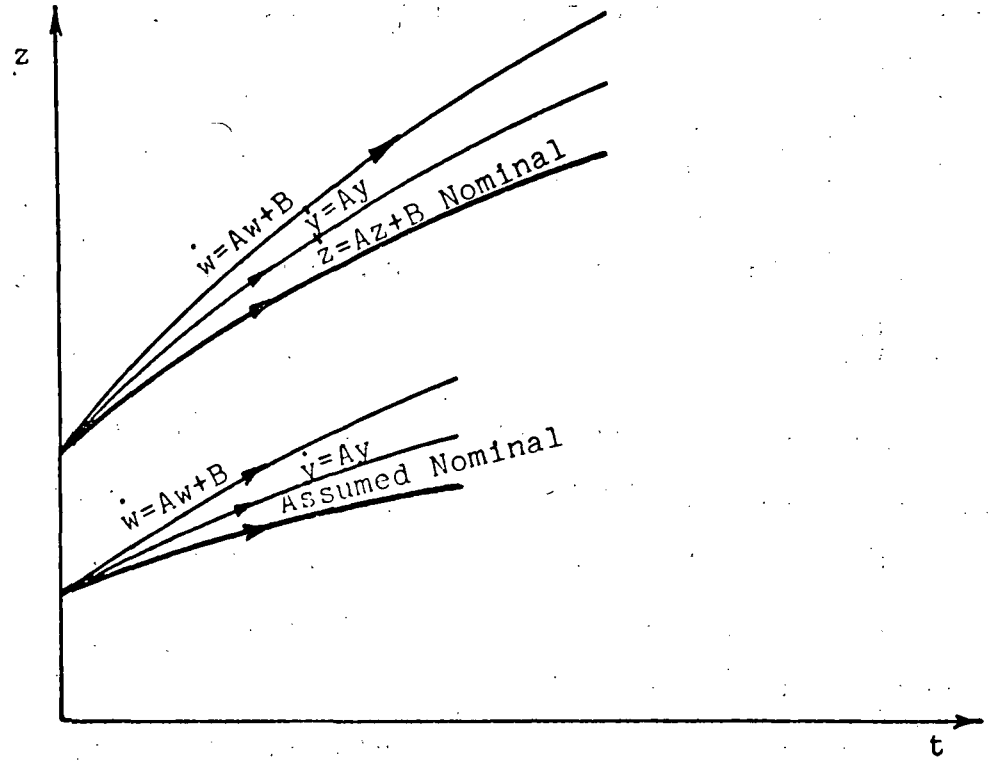
$h(z_f,t_f)$, to a change in the terminal values of z_f itself. The right hand side of Eq. (4.13) is the variation of this general set of terminal boundary conditions $\delta h(t_f)$. A first order expansion of the terminal boundary conditions may be made, $dh = \delta h + \dot{h}dt_f$, and substituted into Eq. (4.13) to yield

$$dh = \left[\frac{\partial h}{\partial z}\right]_f Y(t_0,t_f)C + \dot{h}dt_f \qquad (4.14)$$

where dh is a $2n+1-p$ vector of terminal constraint dissatisfaction, $\left[\frac{\partial h}{\partial z}\right]_f$ is an $2n+1-p \times 2n$ matrix of partial derivatives, $Y(t_0,t_f)$ is an $2n \times 2n-p$ matrix of the terminal values of the homogeneous solutions, C is a $2n-p$ vector of corrections to be determined, $\dot{h}$ is a $2n+1-p$ vector of time rates of change of the terminal constraints and dt_f is the time correction to be determined.

The Eq. (4.14) just derived is analogous to Eq. (3.29) developed for the Method of Perturbation Functions. The major exception is that in the present case the nonlinear differential equations of motion and the Euler-Lagrange equations are linearized. If the optimization problem is reduced to a two-point boundary value problem as discussed in Section 2.2, p becomes equal to n and the implementation of the two methods is similar.

The computational procedure may be followed by referring to an illustration of the Modified Quasilinearization Method (MQM):



(1) Integrate the $2n$ linear nonhomogeneous differential equations, Eq. (4.3), forward from t_0 to t_f with starting conditions consisting of the n known initial conditions and n assumed values for the unknown parameters. The A and B matrices are evaluated from the previous nominal (on the first iteration the assumed nominal is used).

(2) Integrate the 2n linear homogeneous differential equations, Eq. (4.4), forward, simultaneously with the Eq. (4.5), from t_0 to t_f with n starting conditions consisting of a unit perturbation of the variables that corresponds to the unknown initial conditions.

(3) Solve the n+1 linear algebraic equations, Eq. (4.14), for a linear determination of the corrections that must be applied to the assumed initial values and terminal time.

(4) Integrate the 2n linear nonhomogeneous differential equations, Eq. (4.3), forward from t_0 to $t_f + \delta t_f$ with the initial conditions updated by the recently calculated corrections. This integration yields a new nominal.

(5) The process is continued until the metric ρ and the corrections become less than some preselected values.

It should be noted that this approach could have used the adjoint functions rather than the perturbation functions. In this case, its implementation will require the use of a set of equations adjoint to the homogeneous equations, Eq. (4.4), and its development runs parallel to the method

discussed in Section 3.1. The algebraic equation to be solved becomes

$$dh = \Theta'(t_f,t_0)\delta z(t_0) + \dot{h}dt_f \qquad (4.15)$$

where Θ' is an n+1 x 2n matrix resulting from the simultaneous backward integration of the adjoint equations.

4.3 Iteration Philosophy for the Quasilinearization Methods

The iteration scheme for the Quasilinearization Methods simply consist of a procedure to iteratively determine the initial values of the Lagrange multipliers so as to decrease the metric ρ. The control is eliminated from the differential equations, Eq. (4.1), by using the optimality conditions, Eq. (2.25), and the linearized differential equations are integrated during each iteration. Even though the optimality conditions are used, the trajectory iterations do not represent optimal solutions because the trajectories are generated from a linearized version of the nonlinear differential equations. The only remaining requirement is to reduce the metric ρ to an acceptable value, which means that an optimal solution has been converged upon.

With the Method of Generalized Newton-Raphson, the terminal values of the desired variables are introduced and essentially forced to satisfaction on each iteration. The metric ρ is reduced to an acceptable value by iterating on an assumed value of terminal time. Then one of the desired

terminal values is used in a scalar application of the Newton-Raphson method to determine a new terminal time.

Iteration Scheme 1 - This scheme is used with the Method of Generalized Newton-Raphson, and is one which allows a time iteration to be made while the metric ρ is being decreased. This scheme effectively reduces the metric ρ in conjunction with convergence on the desired terminal time.

A detailed procedure of Iteration Scheme 1 follows:

(1) Assume a solution for the $2n$ trajectory variables and a terminal time.

(2) Make one trajectory iteration by integrating forward the homogeneous and nonhomogeneous equations, Eqs. (4.4) and (4.5), respectively. Determine the corrections and integrate the nonhomogeneous equation once again with the new initial conditions. This last integration is considered a new nominal and the metric ρ_1 is determined for this nominal and the assumed trajectory.

(3) Make one more trajectory iteration and obtain a new metric, ρ_2.

(4) Using one of the desired terminal values make a Newton-Raphson iteration to obtain a new value of terminal time.

(5) Make two more trajectory iterations and record the value of the metric ρ_3 .

(6) If the metric ρ_3 is less than the metric ρ_2 , make another time iteration and continue the process.

(7) If the metric ρ_3 is greater than the metric ρ_2 , continue the trajectory iterations until the metric becomes less than ρ_2 . Then make a time iteration and continue the process.

(8) Terminate the procedure when the time corrections and the current metric become less than some pre-selected values.

Iteration Scheme 2 - This scheme is used on the Modified Quasilinearization Method and is similar to Iteration Scheme 1 presented for the Perturbation Methods. When the MGNR is used, the terminal values of the desired variables are introduced in such a manner that a full correction is requested on every iteration. It is expected that if a full correction is requested in cases where the linear representation is poor, the sequence of linear solutions will diverge. The less severe request of only a percentage correction is applied with the Modified Quasilinearization Method and the linear algebraic equation that contains the n+1 corrections is

$$dh = \left[\frac{\partial h}{\partial z}\right]_f Y(t_0,t_f)C + \dot{h}dt_f \quad . \tag{4.16}$$

The terminal dissatisfaction change for a percentage correction is

$$dh = -ch$$

where c is the desired percentage to be corrected, and the iteration factor c may have values in the range $0 \leqslant c \leqslant 1$.

A detailed procedure of Iteration Scheme 2 follows:

(1) Starting values of the iteration factor and the iteration rate factor are selected. Assume a solution for the $2n$ trajectory variables, and a terminal time.

(2) Make one trajectory iteration by integrating forward the homogeneous and nonhomogeneous equations, determining the corrections and integrating the nonhomogeneous equation once again with the new initial conditions and new terminal time. This last integration is considered a new nominal and the metric ρ is determined for this nominal and the assumed trajectory.

(3) If all the corrections and the metric ρ are less than some preselected values, terminate iterations. If any one correction or the metric ρ is

greater than the preselected value continue the process.

(4) Apply the corrections and make another trajectory iteration, obtaining a new metric ρ .

(5) If the new metric is less than the old metric, increase the iteration factor by the value of the iteration rate factor and continue to iterate. Never allow the iteration factor to be less than the value of the iteration rate factor or greater than unity.

CHAPTER 5

GRADIENT METHODS

The general theory of the gradient concept is now both well known and widely used for the approximate solution to trajectory optimization problems. These methods have a common characteristic in that the influence function concept is used to determine how the performance index and/or a combination of the terminal constraint relations are changed as the control variables are changed. Then a control step is taken in the negative gradient direction, i.e. the direction of steepest descent, so as to extremize the performance index while satisfying certain specified terminal constraint relations.

The implementation of the gradient techniques has been widely varied and relatively arbitrary because although the gradient direction is well defined, the proper sized step in control space is not. The convergence properties of the methods are dependent on judicious selection of this step size and the manner in which it is changed, and several efforts have been made to improve the rather slow terminal convergence of the gradient methods. Unfortunately, because of this inherent arbitrariness in the gradient method, a great amount of human intervention is required to select a proper control step size and still avoid violating the linearity constraints imposed

on the problem. In this sense the implementation of the gradient techniques is an art.

The theoretical development of the gradient techniques discussed here may be shown to follow common approaches. The primary difference being the manner in which the terminal constraints are handled and the method of selecting the control step size. The Method of Steepest Descent uses the terminal constraints in the so-called "hard" form, i.e. the constraints are to be satisfied identically. The Modified Method of Steepest Descent uses the terminal constraints in the so-called "soft" form, i.e. the constraints may be only approximately satisfied.

5.1 Method of Steepest Descent

The theoretical development of the Method of Steepest Descent is well known as discussed in References 17 through 21, and is summarized here only to provide background for the iteration scheme modification. It is desired to determine the control program $u(t)$, where u is a m vector, which will yield an extreme value of some performance index

$$\phi = \phi(x_f, t_f) \tag{5.1}$$

subject to the differential equations of motion

$$\dot{x} = f(x,u,t) \tag{5.2}$$

where x is an n vector while satisfying the terminal constraint relations

$$\Psi = \Psi(x_f, t_f) = 0 \tag{5.3}$$

where Ψ is a q vector. One of the desired terminal constraint relations may be used as a stopping condition,

$$\Omega = \Omega(x_f, t_f) = 0 \tag{5.4}$$

The integration process continues until this stopping condition is satisfied. If the differential equations, Eq. (5.2), are linearized about some nominal path, the resulting equations become

$$\delta\dot{x} = F\delta x + G\delta u \tag{5.5}$$

where F and G are $n \times n$ and $n \times m$ matrices of partial derivatives evaluated on the nominal trajectory, respectively.

The equations adjoint to Eq. (5.5) are

$$\dot{\lambda} = -F^T\lambda \tag{5.6}$$

where λ is an n vector of adjoint variables. This equation may be combined with Eq. (5.5) by premultiplying Eq. (5.5) by λ^T and post multiplying the transpose of Eq. (5.6) by δx and adding the equations to yield

$$\frac{d}{dt}(\lambda^T\delta x) = \lambda^T G\delta u \ . \tag{5.7}$$

Integrating Eq. (5.7) from t_0 to t_f yields

$$(\lambda^T \delta x)_f = \int_{t_0}^{t_f} \lambda^T G \delta u dt + (\lambda^T \delta x)_0 \ . \tag{5.8}$$

The boundary conditions on the adjoint variables are arbitrary and may be chosen for convenience. The object now is to determine how initial state variations and integrated control variations influence the terminal values of the performance index, stopping condition and the terminal constraint relations. If, on separate trials, the terminal values of the adjoint variables are set equal to

$$\lambda_\phi^T(t_f) = \left[\frac{\partial \phi}{\partial x}\right]_f \quad \lambda_\psi^T(t_f) = \left[\frac{\partial \psi}{\partial x}\right]_f \quad \lambda_\Omega^T(t_f) = \left[\frac{\partial \Omega}{\partial x}\right]_f \tag{5.9}$$

where λ_ϕ is an n vector, λ_ψ is a nxq matrix and λ_Ω is an n vector, the desired relations are seen to be

$$d\phi = \int_{t_0}^{t_f} \lambda_\phi^T G \delta u dt + (\lambda_\phi^T \delta x)_0 + \dot{\phi} dt_f \tag{5.10}$$

$$d\psi = \int_{t_0}^{t_f} \lambda_\psi^T G \delta u dt + (\lambda_\psi^T \delta x)_0 + \dot{\psi} dt_f \tag{5.11}$$

$$d\Omega = \int_{t_0}^{t_f} \lambda_\Omega^T G \delta u dt + (\lambda_\Omega^T \delta x)_0 + \dot{\Omega} dt_f \tag{5.12}$$

where

$$\dot{\phi} = \left[\frac{\partial \phi}{\partial x}\dot{x} + \frac{\partial \phi}{\partial t}\right]_f \tag{5.13}$$

$$\dot{\psi} = \left[\frac{\partial \psi}{\partial x}\dot{x} + \frac{\partial \psi}{\partial t}\right]_f \tag{5.14}$$

$$\dot{\Omega} = \left[\frac{\partial \Omega}{\partial x}\dot{x} + \frac{\partial \Omega}{\partial t}\right]_f \tag{5.15}$$

and

$$d\phi = [\delta\phi + \dot{\phi}dt]_f \tag{5.16}$$

$$d\psi = [\delta\psi + \dot{\psi}dt]_f \tag{5.17}$$

$$d\Omega = [\delta\Omega + \dot{\Omega}dt]_f \ . \tag{5.18}$$

The approach presented by Bryson and Denham (18) allows for the specification of a requested terminal dissatisfaction improvement and an allowable step size to be taken in control space. The control step size is defined by

$$dS = \int_{t_0}^{t_f} \frac{1}{2}\delta u^T \, W \, \delta u \, dt \tag{5.19}$$

where the step is a weighted quadratic function of the control deviation. The weighting matrix W is included to improve the convergence characteristics by giving more weight to regions of low sensitivity. However, it is often chosen to be the unity matrix because of the lack of knowledge

concerning the region sensitivity. The criteria used for determining the best elements of this matrix are not given and are found through trial and error procedures.

The stopping condition, Eq. (5.4), is to be identically satisfied so Eq. (5.12) is equated to zero. The terminal time variation dt_f is eliminated from Eqs. (5.10) and (5.11) to yield

$$d\phi = \int_{t_0}^{t_f} \lambda_{\phi\Omega}^T G \delta u dt + (\lambda_{\phi\Omega}^T \delta x)_0 \tag{5.20}$$

$$d\psi = \int_{t_0}^{t_f} \lambda_{\psi\Omega}^T G \delta u dt + (\lambda_{\psi\Omega}^T \delta x)_0 \tag{5.21}$$

where

$$\lambda_{\phi\Omega} = \lambda_\phi - \frac{\dot{\phi}}{\dot{\Omega}} \lambda_\Omega \tag{5.22}$$

$$\lambda_{\psi\Omega} = \lambda_\psi - \lambda_\Omega \frac{\dot{\psi}^T}{\dot{\Omega}} \tag{5.23}$$

The total variation in the performance index due to initial state variations and integrated control variations may be expressed as

$$d\phi = \int_{t_0}^{t_f} \lambda_{\phi\Omega}^T G \delta u dt + (\lambda_{\psi\Omega}^T \delta x)_0 + \nu^T \left[d\psi - \int_{t_0}^{t_f} \lambda_{\psi\Omega}^T G \delta u dt - (\lambda_{\psi\Omega}^T \delta x)_0 \right] + \mu \left[dS - \int_{t_0}^{t_f} \frac{1}{2} \delta u^T W \delta u dt \right] \tag{5.24}$$

where the terminal constraint and the control step relations are adjoined by the use of the ν^T and μ Lagrange multipliers, respectively. The multiplier ν is a q vector and μ is a scalar constant. Since it is desired to determine the control variation which corresponds to the maximum change in the performance index, the first variation of Eq. (5.24) must vanish

$$\delta(d\phi) = \int_{t_0}^{t_f} (\lambda_{\phi\Omega}^T G - \nu^T \lambda_{\Psi\Omega}^T G - \mu \delta u^T W)\delta^2 u dt = 0 \quad . \quad (5.25)$$

This implies that the desired control variation is

$$\delta u = \frac{1}{\mu} W^{-1} G^T (\lambda_{\phi\Omega} - \lambda_{\Psi\Omega}\nu) \quad . \quad (5.26)$$

When this equation is substituted back into Eqs. (5.19) and (5.21) the values of ν and μ are determined as

$$\nu = -\mu I_{\Psi\Psi}^{-1} d\beta + I_{\Psi\Psi}^{-1} I_{\Psi\phi} \quad (5.27)$$

$$\mu = \pm \left[\frac{I_{\phi\phi} - I_{\Psi\phi}^T I_{\Psi\Psi}^{-1} I_{\Psi\phi}}{dS - d\beta^T I_{\Psi\Psi}^{-1} d\beta} \right]^{1/2}$$

where

$$d\beta = d\Psi - (\lambda_{\Psi\Omega}^T \delta x)_0 \quad (5.29)$$

$$I_{\Psi\Psi} = \int_{t_0}^{t_f} \lambda_{\Psi\Omega}^T G W^{-1} G^T \lambda_{\Psi\Omega} dt \quad (5.30)$$

$$I_{\psi\phi} = \int_{t_0}^{t_f} \lambda_{\psi\Omega}^T G W^{-1} G^T \lambda_{\phi\Omega} dt \tag{5.31}$$

$$I_{\phi\phi} = \int_{t_0}^{t_f} \lambda_{\phi\Omega}^T G W^{-1} G^T \lambda_{\phi\Omega} dt \tag{5.32}$$

and $I_{\psi\psi}$ is q x q matrix, $I_{\psi\phi}$ is a q vector and $I_{\phi\phi}$ is a scalar.

Now, combining Eqs. (5.26) through (5.32) yields the desired control program

$$\delta u = \pm W^{-1} G^T (\lambda_{\phi\Omega} - \lambda_{\psi\Omega} I_{\psi\psi}^{-1} I_{\psi\phi}) \left(\frac{dS - d\beta^T I_{\psi\psi}^{-1} d\beta}{I_{\phi\phi} - I_{\psi\phi}^T I_{\psi\psi}^{-1} I_{\psi\phi}} \right)^{1/2}$$

$$+ W^{-1} G^T \lambda_{\psi\Omega} I_{\psi\psi}^{-1} d\beta \tag{5.33}$$

where the positive sign is used if ϕ is to be maximized and the negative sign used if ϕ is to be minimized. The previous control program is now modified as follows:

$$u_{new} = u_{old} + \delta u .$$

The computational procedure for the Method of Steepest Descent may be summarized by considering the following:

(1) Integrate the n differential equations of motion, Eq. (5.2), forward in time using an assumed control program and the desired initial conditions

for the state variables. This integration is continued until the stopping condition, Eq. (5.4), is satisfied. The value of the state variables are stored at each point in time.

(2) Integrate the n adjoint equations, Eq. (5.6), backward $q+2$ times with the starting conditions, Eq. (5.9). The coefficient matrix F is obtained from the nominal generated on the forward integration.

(3) Integrate the Eqs. (5.30), (5.31) and (5.32) backwards simultaneously with the adjoint equations using zero as initial conditions to yield the values for $I_{\psi\psi}$, $I_{\psi\phi}$, and $I_{\phi\phi}$.

(4) Select a desired improvement in the terminal dissatisfaction $d\Psi$ for the next iteration.

(5) Select a reasonable value for the mean square control deviation from the previous control program by using

$$dS = \frac{1}{2} W \delta u_{ave}^2 (t_f - t_0) .$$

This will provide a value for the control step dS.

(6) Use the selected values of $d\Psi$ and dS to calculate the numerator under the radical in Eq. (5.33). If this quantity is negative, determine the $d\Psi$ that

makes the quantity vanish. It is is positive, use the quantity as it is.

(7) Calculate the δu as given in Eq. (5.33) and alter the assumed control program.

(8) This procedure is continued until the control variations are less than some preselected value.

5.2 Modified Method of Steepest Descent

The theoretical development of the Modified Method of Steepest Descent, which uses the penalty function technique for handling the terminal constraints, is similar to the conventional method discussed in Section 5.1. The primary difference is that the terminal constraint relation is included, in the "soft" form, with the performance index to form a penalty function

$$P(x_f,t_f) = W_0\phi^2(x_f,t_f) + \sum_{i=1}^{q} W_i\Psi_i^2(x_f,t_f) \qquad (5.35)$$

where the W_i's are weighting constants. If these constants are sufficiently large, minimizing the penalty function is essentially the same as minimizing the performance index ϕ and driving the terminal constraints Ψ to zero.

To determine how this penalty function is related to initial state variations and the integration control variations,

the Eq. (5.8) is used. Selecting the terminal boundary condition for the adjoint equations, Eq. (5.6) to be

$$\lambda_P^T(t_f) = \left[\frac{\partial P}{\partial x}\right]_f \tag{5.36}$$

$$\lambda_\Omega^T(t_f) = \left[\frac{\partial \Omega}{\partial x}\right]_f \tag{5.37}$$

where λ_P is an n vector and λ_Ω is a scalar, yields

$$dP = \int_{t_0}^{t_f} \lambda_P^T G\,\delta u dt + (\lambda_P^T \delta x)_0 + \dot{P} dt_f \tag{5.38}$$

$$d\Omega = \int_{t_0}^{t_f} \lambda_\Omega^T G\,\delta u dt + (\lambda_\Omega^T \delta x)_0 + \dot{\Omega} dt_f = 0 \tag{5.39}$$

where

$$\dot{P} = \left[\frac{\partial P}{\partial x}\dot{x} + \frac{\partial P}{\partial t}\right]_f \tag{5.40}$$

$$\dot{\Omega} = \left[\frac{\partial \Omega}{\partial x}\dot{x} + \frac{\partial \Omega}{\partial t}\right]_f \tag{5.41}$$

The stopping condition, Eq. (5.39), must be identically satisfied. Hence dt_f can be determined from Eq. (5.39) and used to eliminate dt_f from Eqs. (5.38). The result can be expressed as

$$dP = \int_{t_0}^{t_f} \lambda_{P\Omega}^T G\,\delta u dt + (\lambda_{P\Omega}\delta x)_0 \tag{5.42}$$

where

$$\lambda_{P\Omega}^{T} = \lambda_{P} - \frac{\dot{P}}{\dot{\Omega}}\lambda_{\Omega} \quad . \tag{5.43}$$

Now, it is desired to determine the control variation which maximizes the change in the penalty function dP . To insure the predominance of first order effects, a control step size constraint is adjoined to the total variation of the penalty function, to obtain

$$dP = \int_{t_0}^{t_f} \lambda_{P\Omega}^{T} G \delta u dt + \mu \left[dS - \int_{t_0}^{t_f} \frac{1}{2} \delta u^{T} \delta u dt \right] + (\lambda_{P\Omega}^{T} \delta x)_{0} \quad . \tag{5.44}$$

If the above Eq. (5.44) is to assume a maximum value, the first variation must vanish, or

$$\delta(dP) = \int_{t_0}^{t_f} (\lambda_{P\Omega}^{T} G - \mu \delta u^{T}) \delta^{2} u dt = 0 \tag{5.45}$$

which implies that

$$\delta u = K G^{T} \lambda_{P\Omega} \tag{5.46}$$

where K is a constant equal to $1/\mu$. This expression could be written

$$\delta u = K H_{u}^{T} \tag{5.47}$$

where H is defined as the generalized Hamiltonian, $\lambda_{P\Omega}^{T} G$.

This equation implies that the control variation which maximizes the penalty function change is proportional to the magnitude of the control gradient and in either the positive or negative gradient direction, depending on the sign of K. The constant K may be interpreted as the control step size in the gradient direction. When the gradient H_u approaches zero, the control variation also vanishes.

The penalty function change is evaluated by substituting Eq. (5.47) into Eq. (5.42) to yield

$$dP = K \int_{t_0}^{t_f} H_u H_u^T \, dt \; . \tag{5.48}$$

The computational procedure for the Modified Method of Steepest Descent may be summarized by considering the following:

(1) Integrate the n differential equations of motion, Eq. (5.2), using an assumed control program and the desired initial conditions of state. This integration is continued until the stopping condition, Eq. (5.4), is satisfied.

(2) Integrate the n adjoint equations, Eq. (5.6), backward one time with the starting condition, Eq. (5.43), or

$$\lambda^T_{P\Omega}(t_f) = \left[\left(\frac{\partial P}{\partial x}\right) - \left(\frac{\dot{P}}{\dot{\Omega}}\right)\left(\frac{\partial \Omega}{\partial x}\right)\right]_f$$

forming the coefficient F from the nominal path generated on the forward integration.

(3) Having obtained the solution $\lambda^T_{P\Omega}(t)$ the term $H_u = \lambda^T_{P\Omega} G$ may be formed.

(4) The square of $\lambda^T_{P\Omega} G$ may be integrated from t_0 to t_f. Then, using Eq. (5.48), the step size K may be determined by specifying a desired penalty function change dP.

(5) The control variation may be determined from Eq. (5.47) and applied to the assumed control program.

(6) The procedure continues until the penalty function reaches a minimum.

It must be noted that the specified penalty function change, and hence the step size K is arbitrary, and the judicious selection of K becomes a key factor in increasing the convergence rate. An automatic procedure for its selection is desired.

5.3 Iteration Philosophy for the Gradient Methods

The iteration schemes for the gradient methods simply consist of a procedure to iteratively determine a control program so as to extremize a performance index while simultaneously driving the terminal constraint dissatisfaction to zero. The nonlinear differential equations of motion are integrated during each iteration, and the adjoint equations are used to determine how the variation of different terminal quantities are influenced by initial state variations and integrated control variations. The optimality condition, $H_u = 0$, is not used in the formulation, and hence is never identically satisfied.

A minimization of performance index requires a control step to be taken in the negative gradient direction, consistent with the specified terminal constraints, but the size of this step is not defined by considering the theoretical development of the gradient technique itself. Hence, the most severe disadvantage of these techniques is the arbitrariness. Usually a satisfactory convergence rate can only be achieved by experienced personnel.

A primary objective of the present study is to develop an iterative scheme that removes some of the arbitrariness, and increases the convergence rate. Since the weighting matrix W, introduced in Eq. (5.19) is arbitrary, some rational basis for its selection is needed. This problem is approached by

examining an integral form of the Weierstrass E-Function which approximates the change in the performance index or the penalty function. This change is approximated by

$$dP^* \simeq \int_{t_0}^{t_f} E(x^*, \dot{x}^*, \dot{x}, t)dt \qquad (5.49)$$

where E is the Weierstrass E-Function as developed by Gelfand and Fomin (27). The E-Function is defined as

$$E = F(x^*,\dot{x},t) - F(x^*,\dot{x}^*,t) - \frac{\partial F}{\partial \dot{x}^*}(x^*,\dot{x}^*,t)(\dot{x}-\dot{x}^*) \qquad (5.50)$$

and for the system being considered

$$F(x,\dot{x},t) = H(x,u,t) - \lambda^T\dot{x} \qquad (5.51)$$

where $H = \lambda^T f$. The asterisks refer to the optimal path, and the absence of asterisks refer to any nearby path. From the calculus of variations a necessary condition for the existence of a minimum value of performance index is that E be non-negative during the interval $t_0 \leqslant t \leqslant t_f$.

It is noted, by examining Eq. (5.2), that a variation in control is accompanied by a variation in $\dot{x}$, and that a state variation will occur only after a finite duration of time. Hence, the expansion of Eq. (5.49) is made by considering that the control deviation is not accompanied by a change in state. The Eq. (5.49) is now written

$$dP^* \simeq \int_{t_0}^{t_f} (H - H^*)dt \ . \tag{5.52}$$

The first term of the integrand may be expanded in a Taylor's series about the optimal path at each point in time

$$H \simeq H^* + H_u{}^*\delta u + \tfrac{1}{2}\delta u^T H_{uu}{}^*\delta u + \ldots \tag{5.53}$$

and substituting the above equation into Eq. (5.52) and recalling that $H_u{}^* = 0$ on the optimal path results in

$$dP^* \simeq \int_{t_0}^{t_f} \tfrac{1}{2}\delta u^T H_{uu}{}^*\delta u dt \ . \tag{5.54}$$

This equation represents the deviation in the performance index associated with the deviation of the control program from an optimal control program. It must be stated that $H_{uu}{}^*$ is not known until the optimal trajectory is converged upon, but the expression, Eq. (5.54), becomes increasingly accurate as convergence progresses.

An expression identical to Eq. (5.54) may be derived for the performance index change by considering the second variation of the functional I as presented in Eq. (2.5). This approach requires that the control variations are not accompanied by state deviations and that an optimal trajectory is used as the reference path.

The term H_{uu}^{*} is approximated by using the generalized Hamiltonian and the optimality condition, and may be derived as

$$H_{uu}^{*} \simeq \frac{T}{m}\sqrt{\lambda_1^2 + \lambda_2^2} \qquad (5.55)$$

for the Earth-Mars transfer and the Earth launch examples discussed in Appendix A.2.

The Eq. (5.54) indicates that the performance index increase is approximately equal to the integral of a weighted quadratic form of the control deviation, where the weighting is given by H_{uu}^{*}. This same quadratic form appears in Eq. (5.19) for the Method of Steepest Descent, except the weighting matrix W is undefined. This matrix was introduced to provide different weights to control regions of different sensitivity, and may still be used to restrict the control step size. The Eq. (5.19) is then introduced into an expression for the performance index increase as shown in Eq. (5.24). Hence, it is reasonable to interpret the weighting matrix to be H_{uu}^{*}, thus becoming an easily determined specified matrix.

Iteration Scheme 1 - The first iteration scheme for the Method of Steepest Descent follows the procedure outlined in Section 5.1. The weighting matrix W is set equal to the unity matrix, and hence the control variations at all points in time are given the same weight.

Iteration Scheme 2 - The second iteration scheme for the Method of Steepest Descent also follows the procedure outlined in Section 5.1. However, the weighting matrix W is set equal to H_{uu}^*, and hence the control variation is influenced by a time dependent weighting matrix. The only procedural exception is the one associated with determining the H_{uu}^* matrix.

One of the inaccuracies introduced in the above analysis is that the H_{uu}^* matrix must be evaluated with current trajectory information, rather than the desired optimal values. This problem is eliminated in the Modified Method of Steepest Descent by making the Taylor's expansion about the current nominal trajectory. This expansion results in

$$H^* \simeq H + H_u \delta u + \frac{1}{2}\delta u^T H_{uu} \delta u + ... \qquad (5.56)$$

When this equation is substituted into Eq. (5.52), the relationship for the penalty function change becomes

$$dP^* \simeq \int_{t_0}^{t_f} - (H_u \delta u + \frac{1}{2}\, \delta u^T H_{uu} \delta u) dt \ . \qquad (5.57)$$

The negative sign is now present because the control deviation is toward the optimal, instead of away from it as before.

It is desired for the penalty function change to be extremized, and a necessary condition for this to occur is that the first variation of dP^* vanish. The first variation of Eq. (5.57) is set equal to zero

$$\delta(dP^*) \simeq \int_{t_0}^{t_f} - (H_u + \delta u^T H_{uu})\delta^2 u dt = 0 \ . \qquad (5.58)$$

This implies that

$$\delta u = -H_{uu}^{-1} H_u^T \qquad (5.59)$$

where H_u and H_{uu} are evaluated with current trajectory information. This equation implies the optimal control is in the negative gradient direction, weighted by H_{uu}^{-1} . The approximations involved become increasingly accurate as the convergence process approaches the optimal. It is in this near optimal region that the gradient technique is most deficient, and it is expected that the control law, Eq. (5.59), will assist in nullifying the inherent slowness of convergence. By comparing Eqs. (5.47) and (5.59), it is seen that the gradient step now becomes time dependent, where $K = -H_{uu}^{-1}$, and may be easily calculated on each iteration.

Iteration Scheme 1 - The first iteration scheme associated with the Modified Method of Steepest Descent requires the gradient step determination to be made by using Eq. (5.48). This equation will yield a gradient step after performing the indicated integration and specifying a desired improvement in the penalty function. Caution must be exercised so as not to request such a large penalty function improvement that the linearity assumptions are violated.

A detailed procedure of Iteration Scheme 1 follows:

(1) Integrate the nonlinear differential equations of motion, Eq. (5.2), forward from t_0 to the t_f which satisfies the stopping condition, Eq. (5.4). The desired initial conditions and an assumed control program are used. An initial evaluation of the penalty function P_0 is made.

(2) Integrate the adjoint equations, Eq. (5.6), backwards from t_f using the variables from the forward integration to evaluate the coefficients. The starting conditions are determined by evaluating Eq. (5.43) at the terminal time and are used to generate the solution $\lambda^T_{P\Omega}(t)$.

(3) Having obtained the solution $\lambda^T_{P\Omega}(t)$, the quantity $H_u = \lambda^T_{P\Omega} G$ may be evaluated

(4) The square of H_u may be integrated from t_0 to t_f and using Eq. (5.48), K may be determined by specifying a desired change in the penalty function.

(5) This step size K is used to modify the control variation as stated in Eq. (5.47), and a new control program is determined.

(6) This new control program is used to generate a new nominal and the procedure is repeated.

Iteration Scheme 2 - The second iteration scheme associated with the Modified Method of Steepest Descent is similar to a technique used by Wagner and Jazwinski (21). This scheme involves making three trial forward integrations using different but constant gradient step sizes, and recording the three resulting penalty function values. A second order polynomial is fitted through these points and the step size that corresponds to the minimum value of the penalty function is selected. This method takes full advantage of each adjoint integration by selecting an optimal step size for that iteration.

A detailed procedure of Iteration Scheme 2 follows:

(1) Integrate the nonlinear differential equations of motion, Eq. (5.2), forward from t_0 to the t_f which satisfies the stopping condition, Eq. (5.4). The desired initial conditions and an assumed control program is used. An initial evaluation of the penalty function P_0 is made.

(2) Integrate the adjoint equations, Eq. (5.6), backwards from t_f using the variables from the forward

integration to evaluate the coefficients. The starting conditions are determined by evaluating Eq. (5.43) at the terminal time and are used to generate the solution $\lambda^T_{P\Omega}(t)$.

(3) Having obtained the solution $\lambda^T_{P\Omega}(t)$, the quantity $H_u = \lambda^T_{P\Omega} G$ may be evaluated.

(4) The square of H_u may be integrated from t_0 to t_f and using Eq. (5.48) K_1 may be determined by specifying a desired change in the penalty function.

(5) This step size K_1 is used to modify the control variation as stated in Eq. (5.47), and a new control program is determined.

(6) Integrate the differential equations of motion again using the new control program and record the associated penalty function P_1 .

(7) Depending on whether P_1 is greater or less than P_0 , the step size K_1 is either halved or doubled, respectively.

(8) The control is modified once again and an integration of the differential equations of motion yield a penalty function P_2 .

(9) A second order polynomial is fitted through the three points, and the step size K_{min} is determined

that corresponds to the minimum value of the penalty function.

(10) The control is modified with this K_{min} and the differential equations are integrated to yield a new nominal trajectory. The penalty function resulting from this integration is used to start the cycle over again.

Iteration Scheme 3 - The third iteration scheme associated with the Modified Method of Steepest Descent requires reference to the results given in Eq. (5.59). The implementation of this scheme is extremely simple compared to the first iteration scheme, because no trial forward integrations are required. The time dependent matrix H_{uu}, which may be formed as the adjoint equations are integrated backwards, is easily determined. The control variation for the next iteration is then determined as the H_{uu} matrix is formed.

A detailed procedure of Iteration Scheme 3 follows:

(1) Integrate the nonlinear differential equations of motion, Eq. (5.2), forward from t_0 to the t_f which satisfies the stopping condition, Eq. (5.4). The desired initial conditions and an assumed control program is used for the first iteration. An initial evaluation of the penalty function P_0 is made.

(2) Integrate the adjoint equations, Eq. (5.6), backwards from t_f using the variables from the forward integration to evaluate the coefficients. The starting conditions are determined by evaluating Eq. (5.43) at the terminal time and are used to generate the solution $\lambda^T_{P\Omega}(t)$.

(3) Having obtained the solution $\lambda^T_{P\Omega}(t)$, the quantities H_u and H_{uu} may be evaluated, hence the control modification, Eq. (5.59), may be determined.

(4) The previous control program can be modified and the process continued.

CHAPTER 6

COMPARISON AND DISCUSSION OF THE OPTIMIZATION METHODS AND ITERATION SCHEMES

A meaningful comparison of the optimization methods and associated iteration schemes is extremely difficult to make. One primary reason for this difficulty is that most methods are highly problem dependent, i.e., the characteristics of each method are different for each problem attacked. Furthermore, difficulties arise even if a comparison is made between the optimization methods based on the same physical problem. As an example, suppose it is desired to compare the convergence times of several optimization methods. It is obvious that the convergence time is highly dependent on the integration step size selected, and therefore some reasonable criteria for this selection must be established.

The comparison of the optimization methods and iteration schemes on a numerical basis requires a realistic and representative trajectory problem. The example chosen is a spacecraft moving under the influence of thrust in an inverse square gravitational force field. Specifically, the problems investigated are (1) a low thrust transfer trajectory from Earth to Mars, and (2) an atmospheric Earth launch to circular orbit trajectory. A more detailed discussion of the specific applications is made in Appendix A.2. The time histories of the variables and control

programs that correspond to the optimal trajectories are shown in Figures A.2.1 through A.2.4.

6.1 Selection of Methods for Comparative Study

The trajectory optimization problem has been shown to be theoretically solvable by using several different indirect and direct methods. Of the methods, presented in Chapters 3, 4, and 5, several different approaches are discussed. Some of the more promising methods and associated iteration schemes were selected for computational investigation.

The methods selected for computational investigation are referred to by the following abbreviated names. These methods are:

(1) Method of Adjoint Functions (MAF) - the third approach discussed in Section 3.1.

(2) Method of Perturbation Functions (MPF) - the third approach discussed in Section 3.2.

(3) Method of Generalized Newton-Raphson (MGNR) - the first approach discussed in Section 4.1.

(4) Modified Method of Generalized Newton-Raphson (MMGNR) - the second approach discussed in Section 4.1.

(5) Modified Quasilinearization Method (MQM) - the approach discussed in Section 4.2.

(6) Method of Steepest Descent (MSD) - the approach discussed in Section 5.1.

(7) Modified Method of Steepest Descent (MMSD) - the approach discussed in Section 5.2.

The constants used in the numerical study are given in Appendix A.3.

6.2 Basis of Comparison

A basis of comparison must be established for the comparative study of the optimization methods selected in Section 6.1. The comparison is to be made not only between optimization methods, but between the associated iteration schemes as well.

In a general sense, the following items are considered a basis for comparison for the optimization methods:

(1) Required formulation, application and programming complexity.

(2) Required amount of computer logic and storage.

(3) Ease of use by inexperienced personnel.

(4) Required programming effort for solving different problems.

(5) Effectiveness in solving different problems.

(6) Sensitivity of the convergence characteristics to initially assumed parameters.

(7) Resulting time for convergence.

The iteration schemes are not only concerned with the above items but with the following items as well:

(1) Effectiveness of decreasing the sensitivity of the convergence characteristics of the method to initially assumed parameters.

(2) Effectiveness of decreasing the time for convergence.

6.3 Perturbation Methods

The comparison and discussion of the Perturbation Methods will consist of two separate analyses. The Method of Adjoint Functions, including the normal procedure and Iteration Schemes 1 and 2, is discussed first. The Method of Perturbation Functions with Iteration Scheme 1 is discussed last. The discussion content will include the applicable items listed in Section 6.2.

6.3.1 Method of Adjoint Functions

The required formulation of the Method of Adjoint Functions as discussed in Section 3.1 is simple and straightforward. A general discussion of the applications is presented in Appendix A.2 and a specific application of the MAF is made in Appendix A.2.1. The examples chosen are described by four, first order, nonlinear differential equations of motion, i.e., Newton's

equations for motion in a plane.

The programming effort requires the forward integration of the four differential equations of motion and the four Euler differential equations. Integration of the differential equation for the rate of change of control, i.e., Eq. (2.22), is not required since the control is easily determined and eliminated from the state and Euler equations. These eight dependent variables and the independent variable are stored in computer memory or on tape at each time step during the forward integration for use in forming the A(z,t) matrix. This requires less storage than if each element of the A(z,t) matrix is stored since this would require 64 quantities to be stored at each time step. The A(z,t) matrix must be formed during the backward integration, but this requires very little additional time.

The backwards integration of the eight adjoint differential equations must be made with four different starting vectors, and hence a large percentage of the computation time is spent in this backward integration. The adjoint equations are linear and it is conceivable that a larger integration step or a variable step could be taken. This, however, requires additional programming complexity to insure that the proper coefficients are being formed from the variables stored during the forward integration.

There is an alternative approach that eliminates the storage problem, and hence becomes attractive for problems of large dimension or for ones that require many integration steps.

This approach is one where the differential equations of motion and the Euler equations are integrated backward simultaneously with the adjoint equations. This does not eliminate the forward integration because the terminal values of the state and Euler variables are required to start the backward integration. The sacrifice to eliminate the storage and magnetic tape problems is made by having to integrate an additional set of equations.

For the numerical investigation made, the former procedure is used which means more programming complexity, but also less computer time required. A constant step size was selected for both the forward and backward integrations.

The computer program that uses the MAF requires two initially assumed Lagrange multipliers and an assumed terminal time. These estimates require a familiarity with the physical problem and, to some degree, experience. The computer program is built such that only the subroutines containing the differential equations of motion, the Euler-Lagrange equations, and the adjoint equations must be changed to solve different problems.

Iteration Scheme 1 requires very little computer logic in addition to the Normal Scheme which just requests 100 percent terminal constraint satisfaction on each iteration. Operation is simply transferred to a subroutine where the iteration factor is altered in accordance with the terminal norm criterion explained in Section 3.3.

Iteration Scheme 2 requires some additional programming and computer storage. Basically, the scheme is such that the

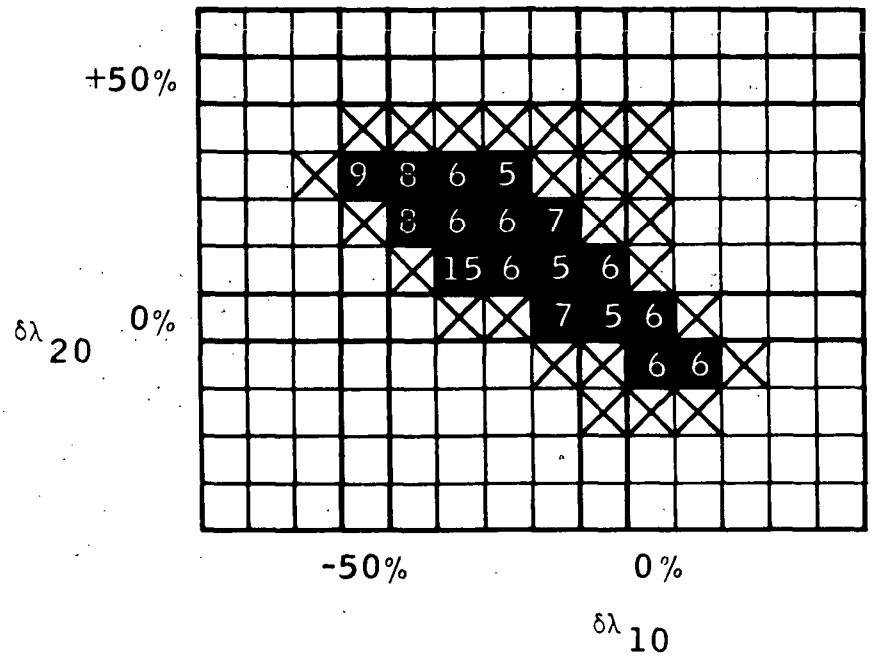
iteration factor is increased, omitting an adjoint integration, until either the terminal constraint norm diverges or a specified number of forward integrations have been made. If the norm does diverge, the last convergent trajectory is used as a nominal, and hence this trajectory must be saved until it is determined whether or not it will be needed. The storage problem can be eliminated, however, by simply regenerating the last convergent trajectory.

The Earth-Mars transfer is completely defined when λ_{10}, λ_{20}, and t_f have been determined, as shown in Appendix A.2.1. The quantity λ_{40} is easily determined to be zero. In an effort to determine how sensitive the method is to poor initial assumptions for the above three quantities, many cases are investigated. These numerical results are best illustrated by building envelopes of convergence, the boundary of which represents the last convergent trial. Points beyond this boundary do not result in a convergent solution. The percentage numbers on the axes represent the percent deviation from the values that result in an optimal solution.

The envelopes of convergence for the MAF, using the Normal Iteration Scheme of requesting a 100 percent correction in the terminal constraints regardless of the terminal norm response, are shown in Figures 1, 2, and 3 for the cases of -20, 0 and 20 percent error in terminal time, respectively.

The physical significance of the convergence envelopes is clear when it is realized, by referring to Appendix A.1, that

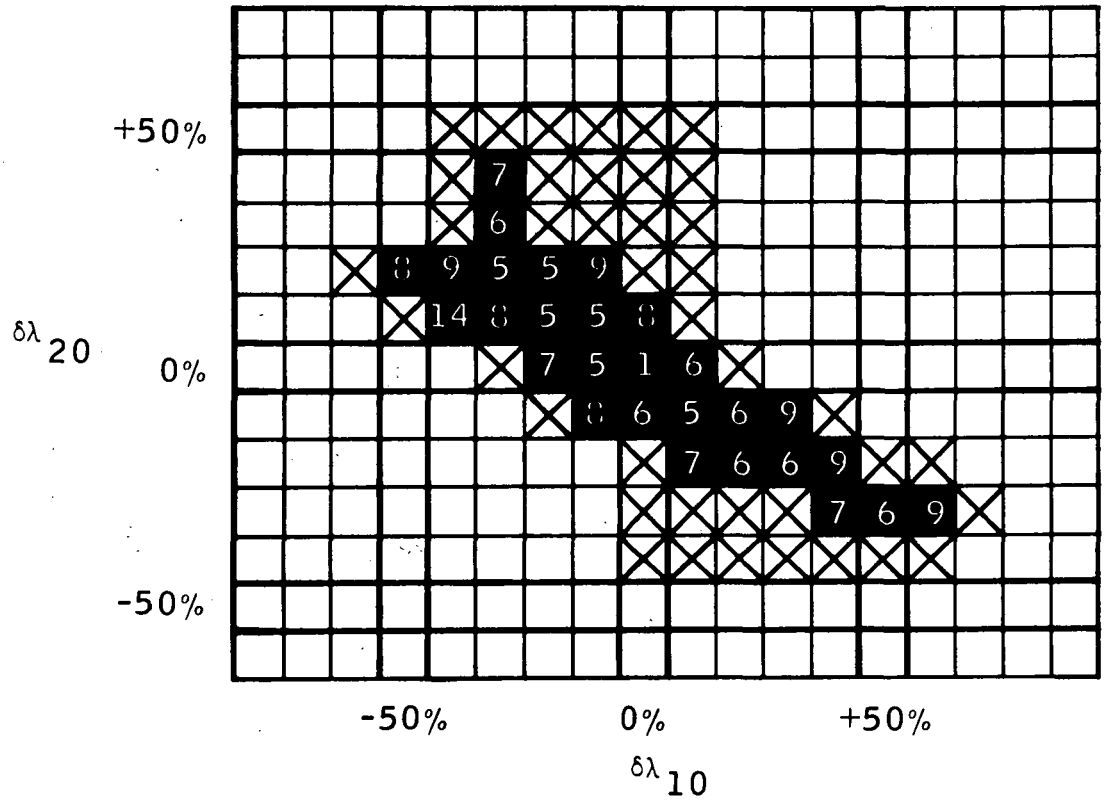
Earth-Mars transfer
Iteration method: MAF
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: -20%



Note: The numbers indicate the iterations required for convergence

Figure 1.- Convergence envelope for the MAF using the normal iteration scheme, initial iteration factor of 100% and terminal time error of -20%.

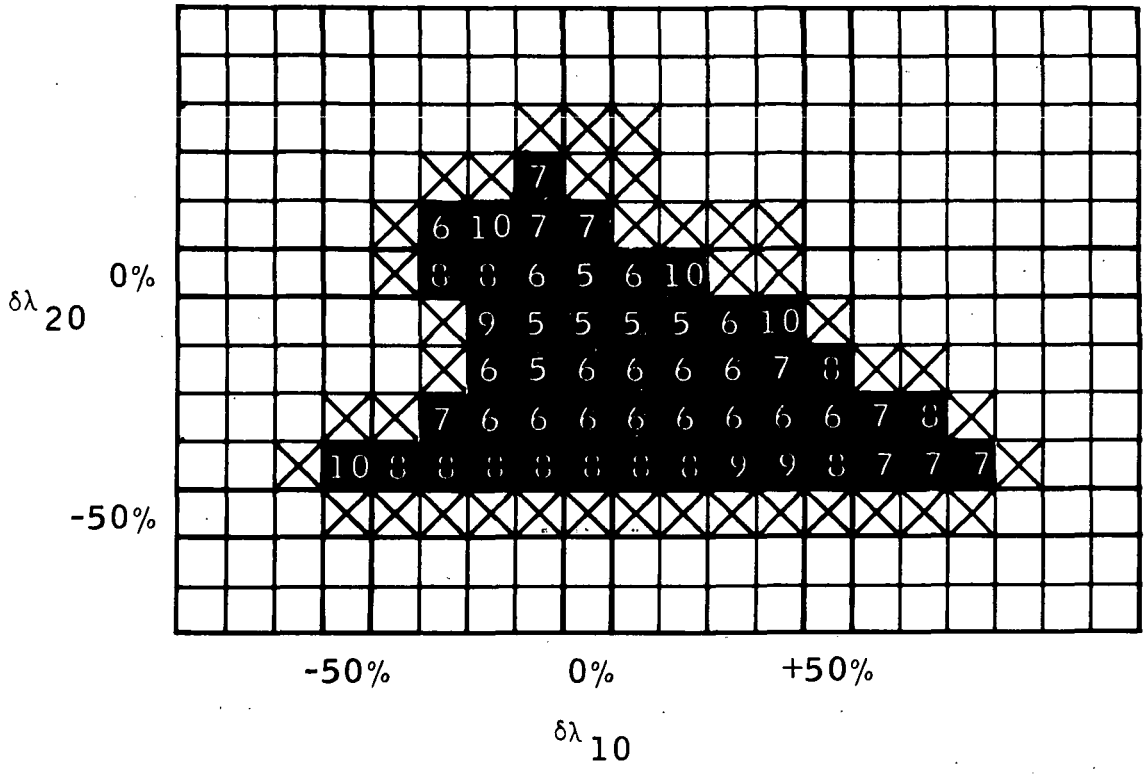
Earth-Mars transfer
Optimization method: MAF
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: 0%



Note: The numbers indicate the iterations required for convergence

Figure 2.- Convergence envelope for the MAF using the normal iteration scheme, initial iteration factor of 100% and terminal time error of 0%.

Earth-Mars transfer
Optimization method: MAF
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: 20%



Note: The numbers indicate the iterations required for convergence

Figure 3.- Convergence envelope for the MAF using the normal iteration scheme, initial iteration factor of 100% and terminal time error of 20%.

the thrust or control angle with respect to the local horizontal is given by $\sin \beta = -\lambda_1/(\lambda_1^2 + \lambda_2^2)^{\frac{1}{2}}$ and $\cos \beta = -\lambda_2/(\lambda_1^2 + \lambda_2^2)^{\frac{1}{2}}$. Points along a 45° diagonal lying in the first and third quadrants represent the optimal initial control angles, but with different values for the individual magnitudes of the Lagrange multipliers. The signs of the initial Lagrange multiplier errors are the same. Points along a 45° diagonal lying in the second and fourth quadrants represent nonoptimal initial control angles for various values in the individual magnitudes of the initial Lagrange multipliers. Down and to the right in the fourth quadrant means the initial control angle is decreasing and up and to the left means the initial control angle is increasing. The signs of the initial Lagrange multiplier errors are opposite.

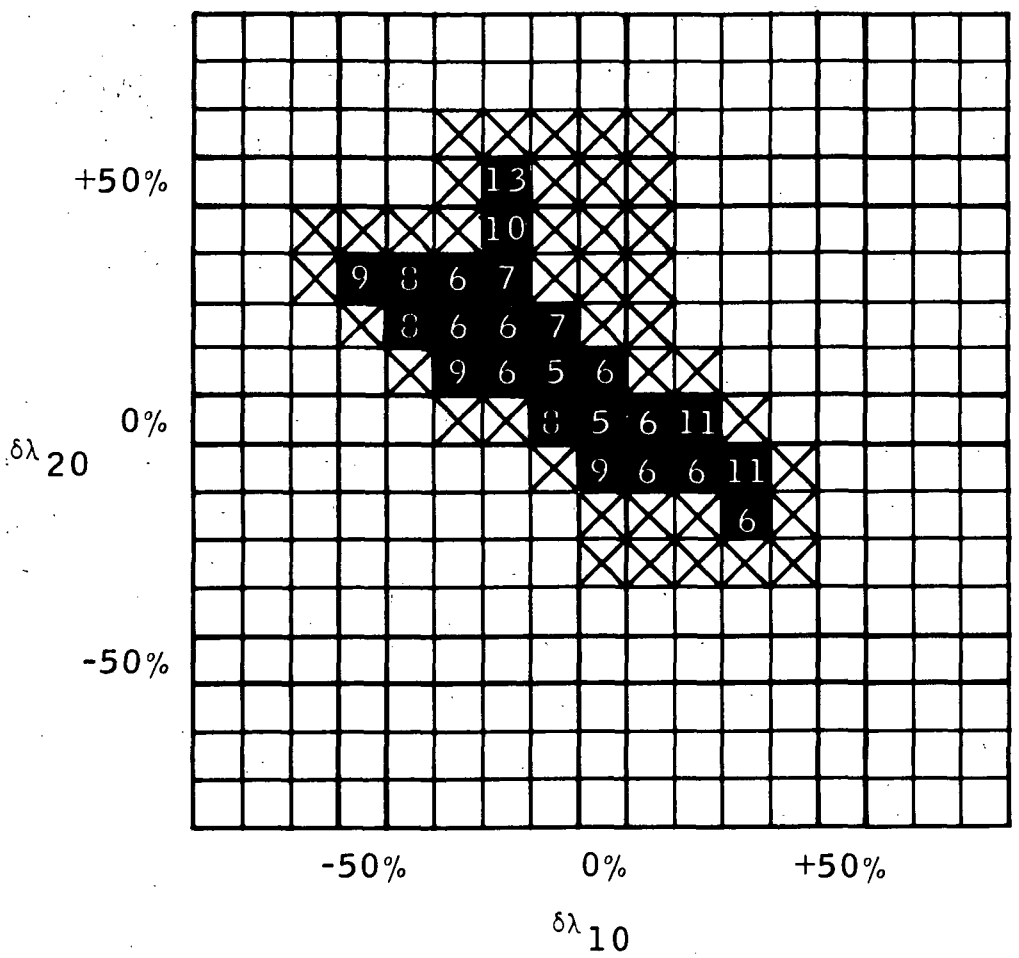
It is seen that the convergent solutions in Figures 1, 2, and 3 remain near the diagonal passing from the second to fourth quadrants. The conclusion must be that for these cases the method is more sensitive to changes in the optimal values of the initial Lagrange multiplier errors that have the same sign, even though the initial control angle remains near optimal for these cases. The method is less sensitive to changes in the initial Lagrange multiplier errors that have the opposite sign, even though the initial control angle is not near optimal. One other interesting characteristic is that as the error in terminal time increases from negative to positive, the envelopes increase in size and move further down into the third and fourth quadrants. The convergence envelope in Figure 2 is approximately

30 percent larger than the one in Figure 1, and the convergence envelope in Figure 3 is approximately 70 percent larger than the one in Figure 2. When a positive terminal time error exists, the method becomes less sensitive to negative λ_{20} errors, but highly sensitive to positive λ_{20} errors.

Iteration Scheme 1, using an initial value for the iteration factor of 100 percent, is effective in increasing the convergence envelope slightly, as illustrated in Figures 4, 5, and 6. These envelopes exhibit the same characteristics as those shown in Figures 1, 2, and 3, except that the envelopes are slightly larger. This increase in size is attributed to the ability of the Iteration Scheme 1 to decrease the iteration factor when the terminal norm diverges. This easement of the requested percentage correction allows some cases to converge when divergence would have occurred had the iteration factor been forced to remain 100 percent for all iterations.

The convergence envelopes are significantly increased by using Iteration Scheme 1 and an initial iteration factor of 50 percent rather than 100 percent. These envelopes are shown in Figures 7, 8, and 9, and are approximately 360, 350 and 260 percent larger, respectively, than the corresponding envelopes for initial iteration factors of 100 percent. The convergent solutions of these envelopes do not remain so near the second to fourth quadrant diagonal as the previous cases although the skewed appearance is still perceptible. One characteristic seen in Figures 4, 5, and 6 becomes more pronounced in Figures 7, 8, and 9 and that is the downward movement of the envelope as the

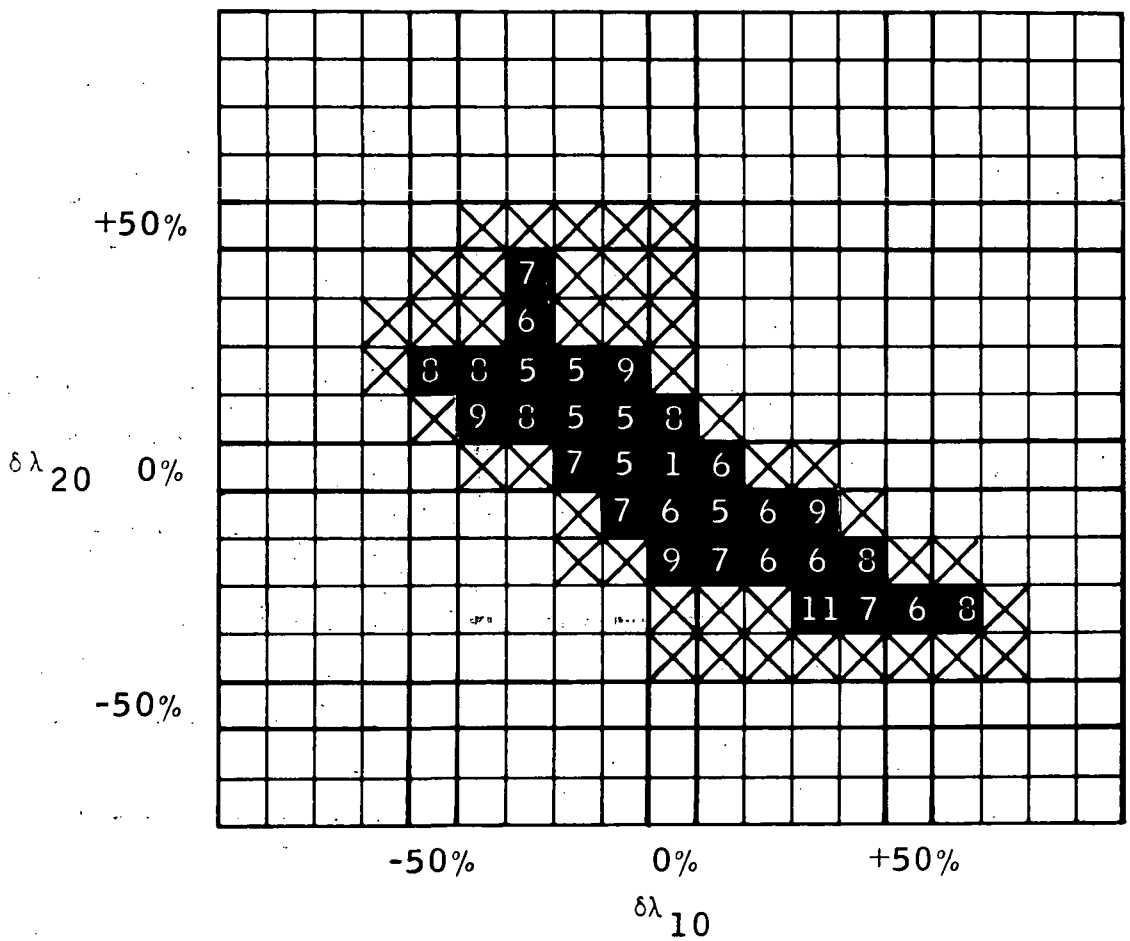
Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 1
Initial iteration factor: 100%
Terminal time error: -20%



Note: The numbers indicate the iterations required for convergence

Figure 4.- Convergence envelope for the MAF using iteration scheme 1, initial iteration factor of 100% and terminal time error of -20%.

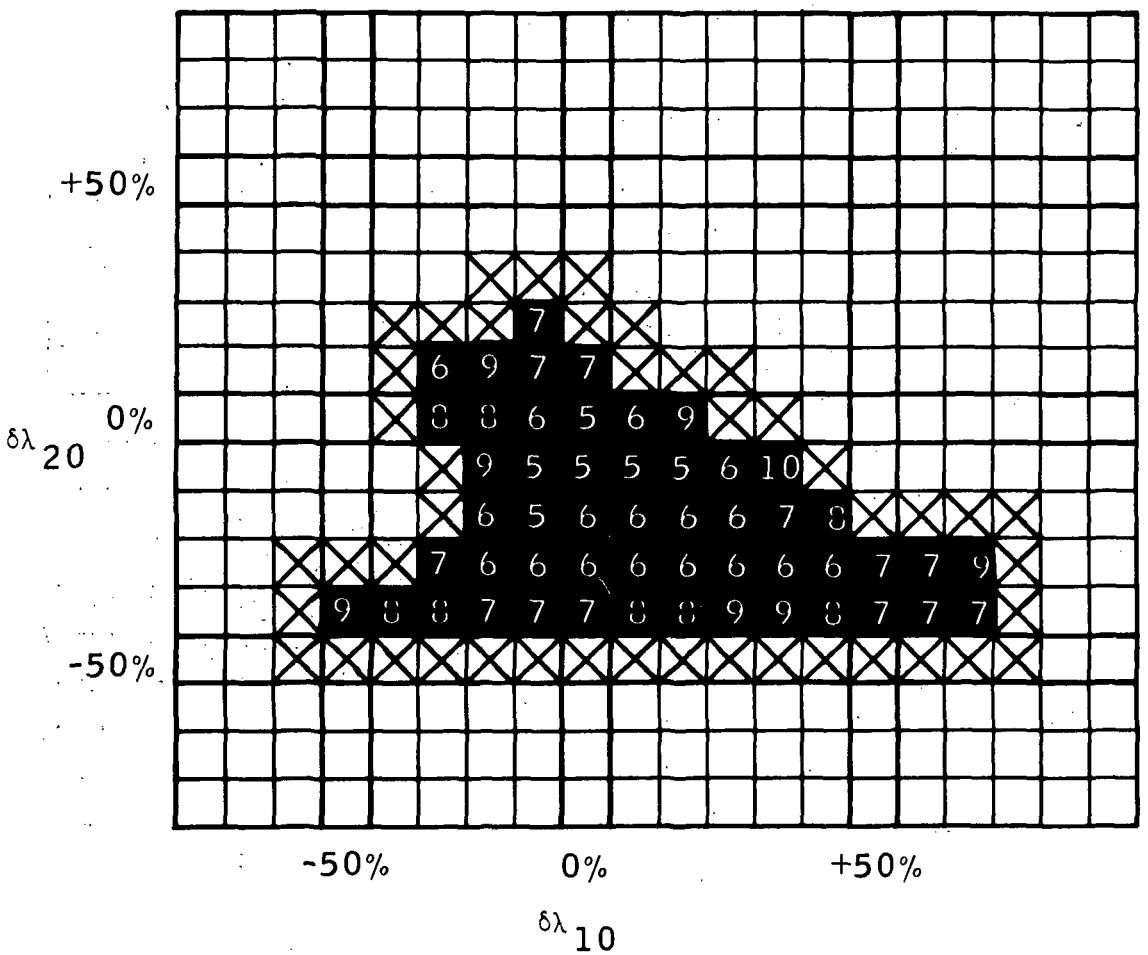
Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 1
Initial iteration factor: 100%
Terminal time error: 0%



Note: The numbers indicate the iterations required for for convergence

Figure 5.- Convergence envelope for the MAF using iteration scheme 1, initial iteration factor of 100% and terminal time error of 0%.

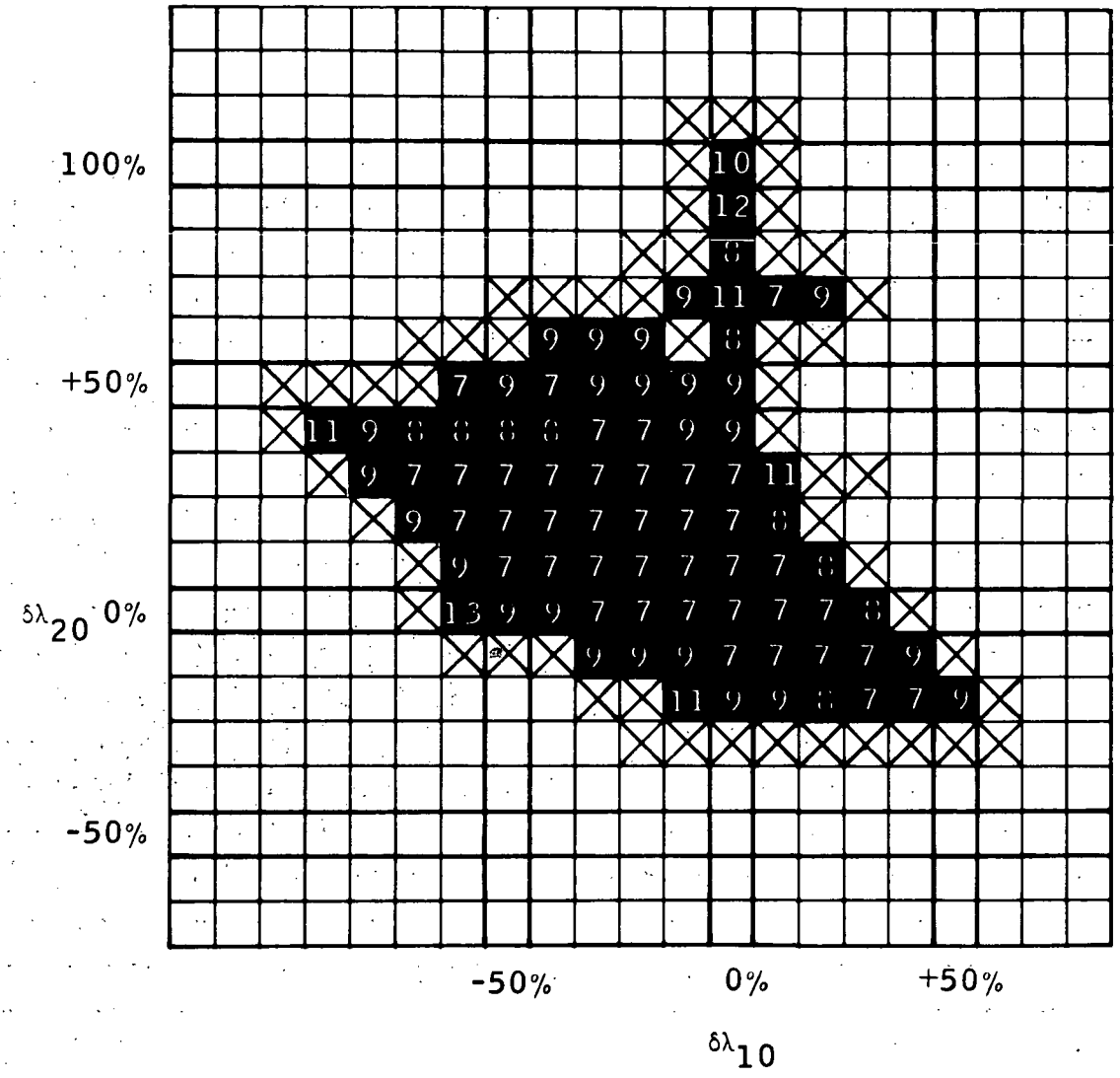
Earth-Mars transfer
Optimization scheme: MAF
Iteration scheme: 1
Initial iteration factor: 100%
Terminal time error: 20%



Note: The numbers indicate the iterations required for convergence

Figure 6.- Convergence envelope for the MAF using iteration scheme 1, initial iteration factor of 100% and terminal time error of 20%.

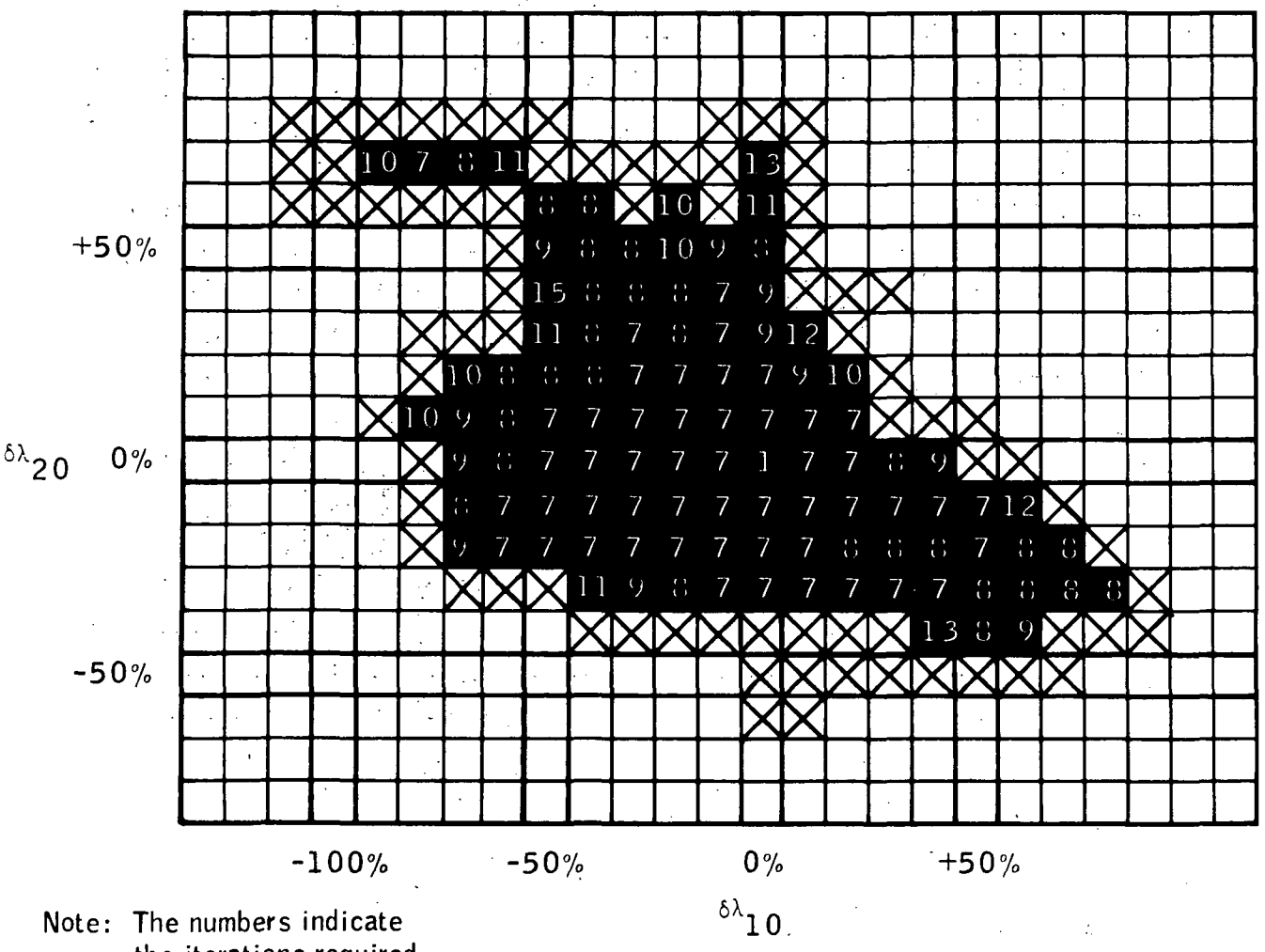
Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 1
Initial iteration factor: 50%
Terminal time error: -20%



Note: The numbers indicate the iterations required for convergence

Figure 7.- Convergence envelope for the MAF using iteration scheme 1, initial iteration factor of 50% and terminal time error of -20%.

Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 1 and 2
Initial iteration factor: 50%
Terminal time error: 0%
Update integer: 1



Note: The numbers indicate the iterations required for convergence

Figure 8.- Convergence envelope for the MAF using iteration schemes 1 and 2, initial iteration factor of 50%, terminal time error of 0% and update integer of 1.

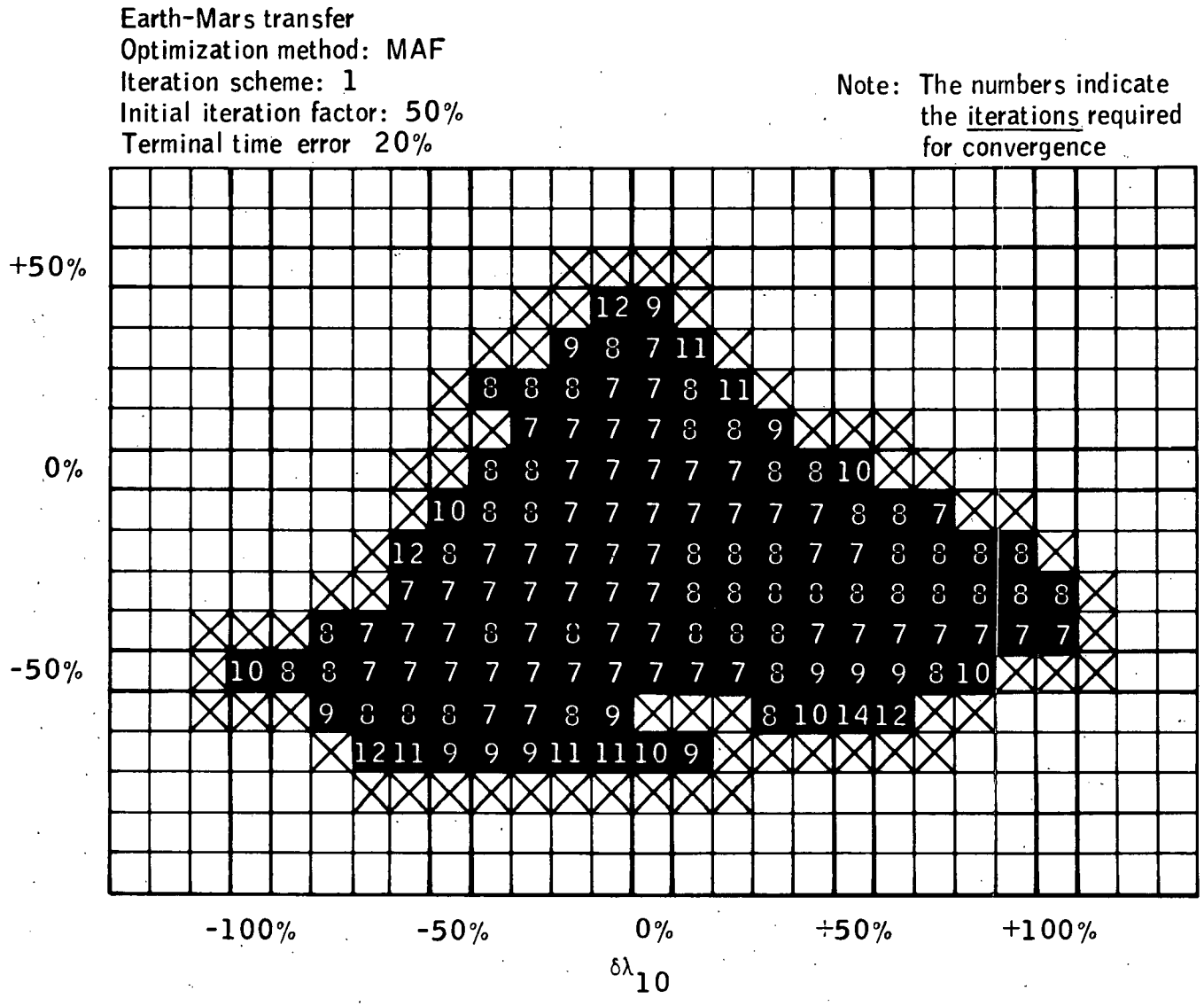


Figure 9.- Convergence envelope for the MAF using iteration scheme 1, initial iteration factor of 50% and terminal time error of 20%.

positive terminal error is increased. This seems reasonable since a negative λ_{20} error, which decreases the initial control angle, combined with a positive t_f error would probably cause the vehicle to intercept Mar's orbit at a low angle. This trajectory would conceivably terminate closer to the optimal point than if the time error were less.

Figures 7, 8, and 9 also display the characteristic that the envelope boundary becomes poorly defined, i.e., more irregular. This emphasizes the fact that many times only a slight numerical difference exists between convergence and divergence, and hence the scheme becomes very unpredictable near the boundaries. This is emphasized further by noting that in many cases a divergence occurs immediately after a relatively low iteration convergence case.

Iteration Scheme 2 continues to integrate the differential equations forward and skips the adjoint equation integration unless a divergence occurs or a specified number (updating integer) of forward passes have been made. Figures 10, 11, and 12 show Iteration Scheme 2 for an initial iteration factor of 50 percent and updating integers of 2, 4, and 6, respectively. The figures indicate the total iterations and the number of adjoint integrations required. Figure 8, showing Iteration Scheme 1, may be considered a special case of Iteration Scheme 2 where the updating integer is unity. A comparison of these figures reveals that no significant change in the convergence envelope size or shape has resulted from the application of Iteration Scheme 2 or

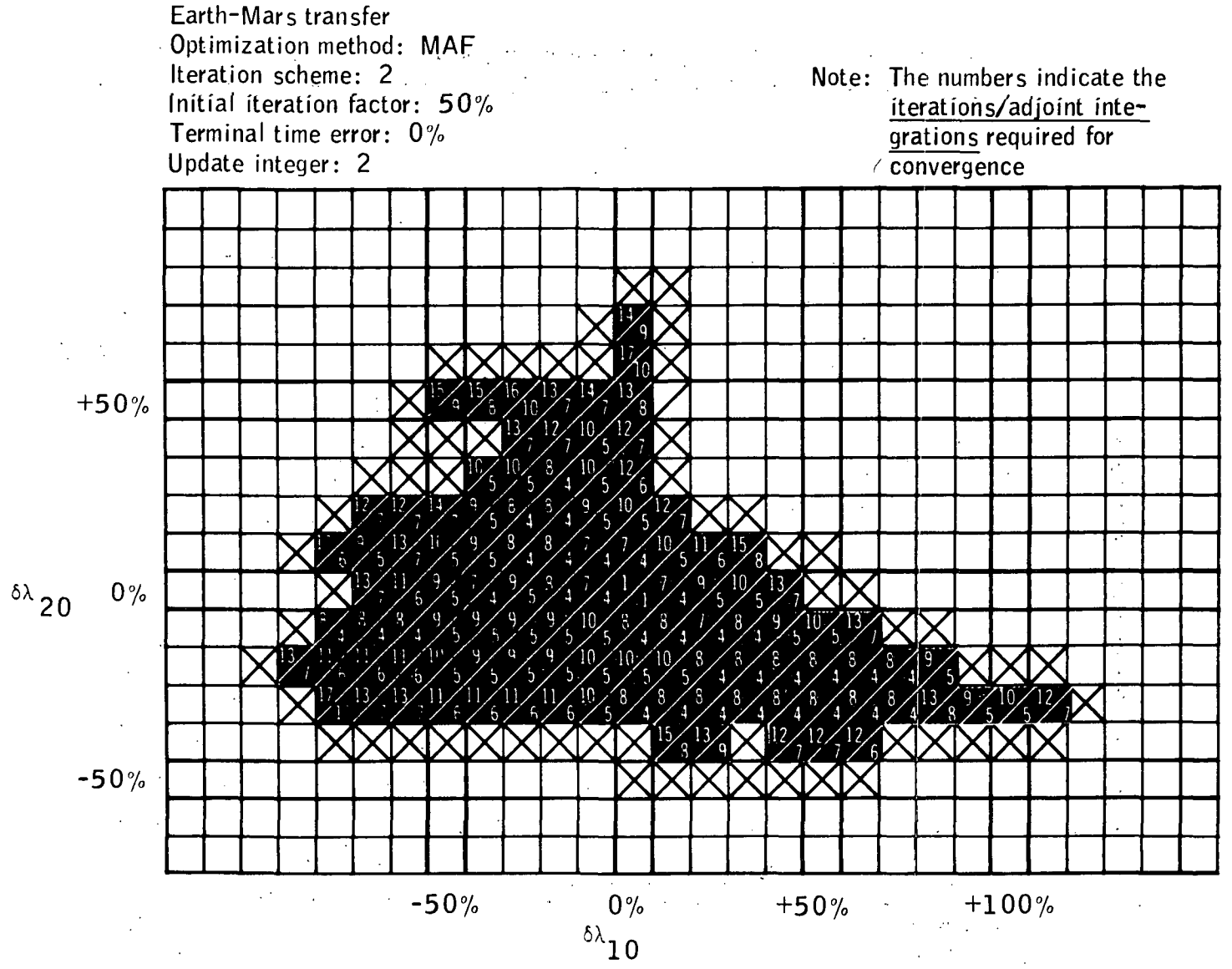


Figure 10.- Convergence envelope for the MAF using iteration scheme 2, initial iteration factor of 50%, terminal time error of 0% and update integer of 2.

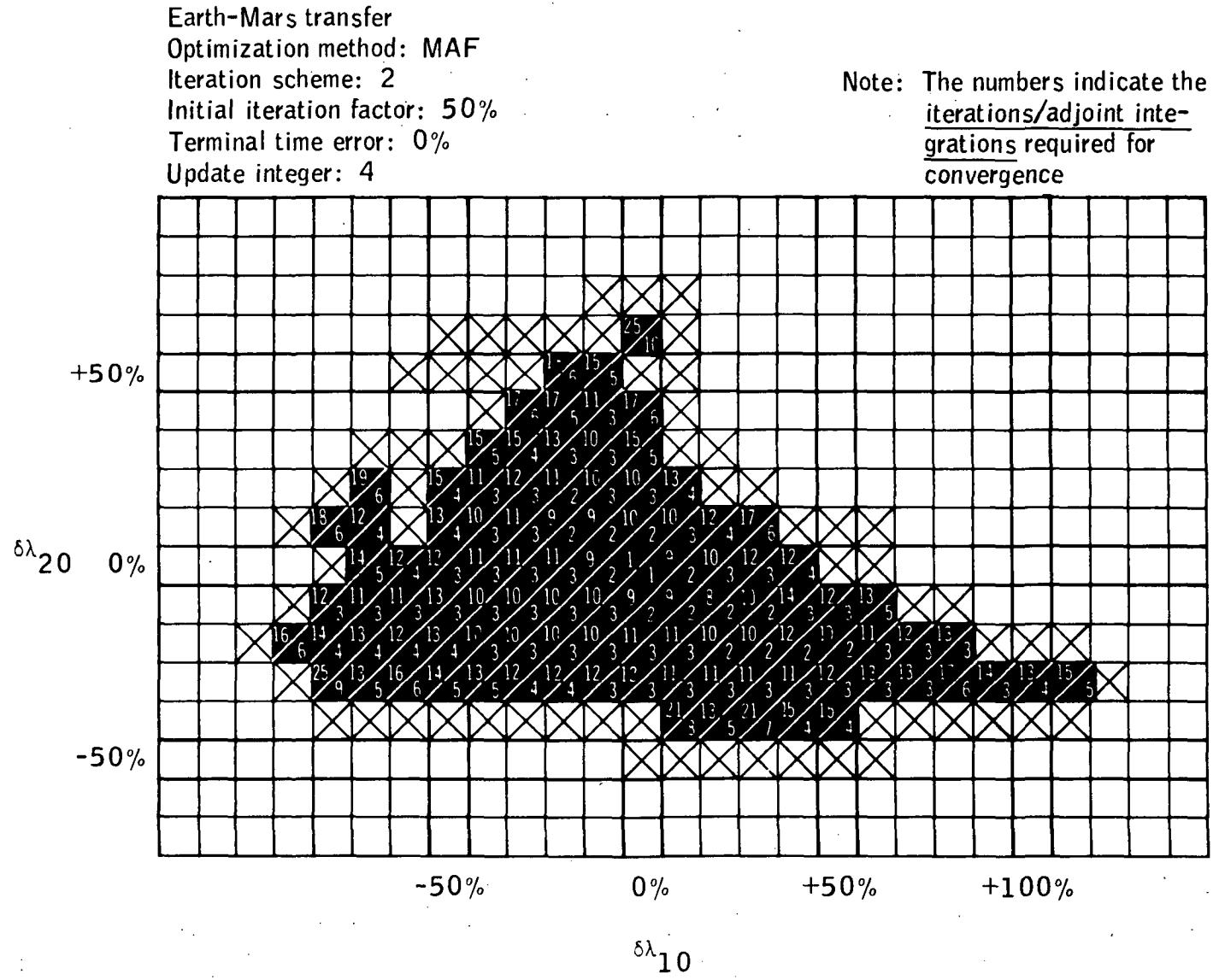


Figure 11.- Convergence envelope for the MAF using iteration scheme 2, initial iteration factor of 50%, terminal time error of 0% and update integer of 4.

Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 2
Initial iteration factor: 50%
Terminal time error: 0%
Update integer: 6

Note: The numbers indicate the iterations/adjoint interations required for convergence

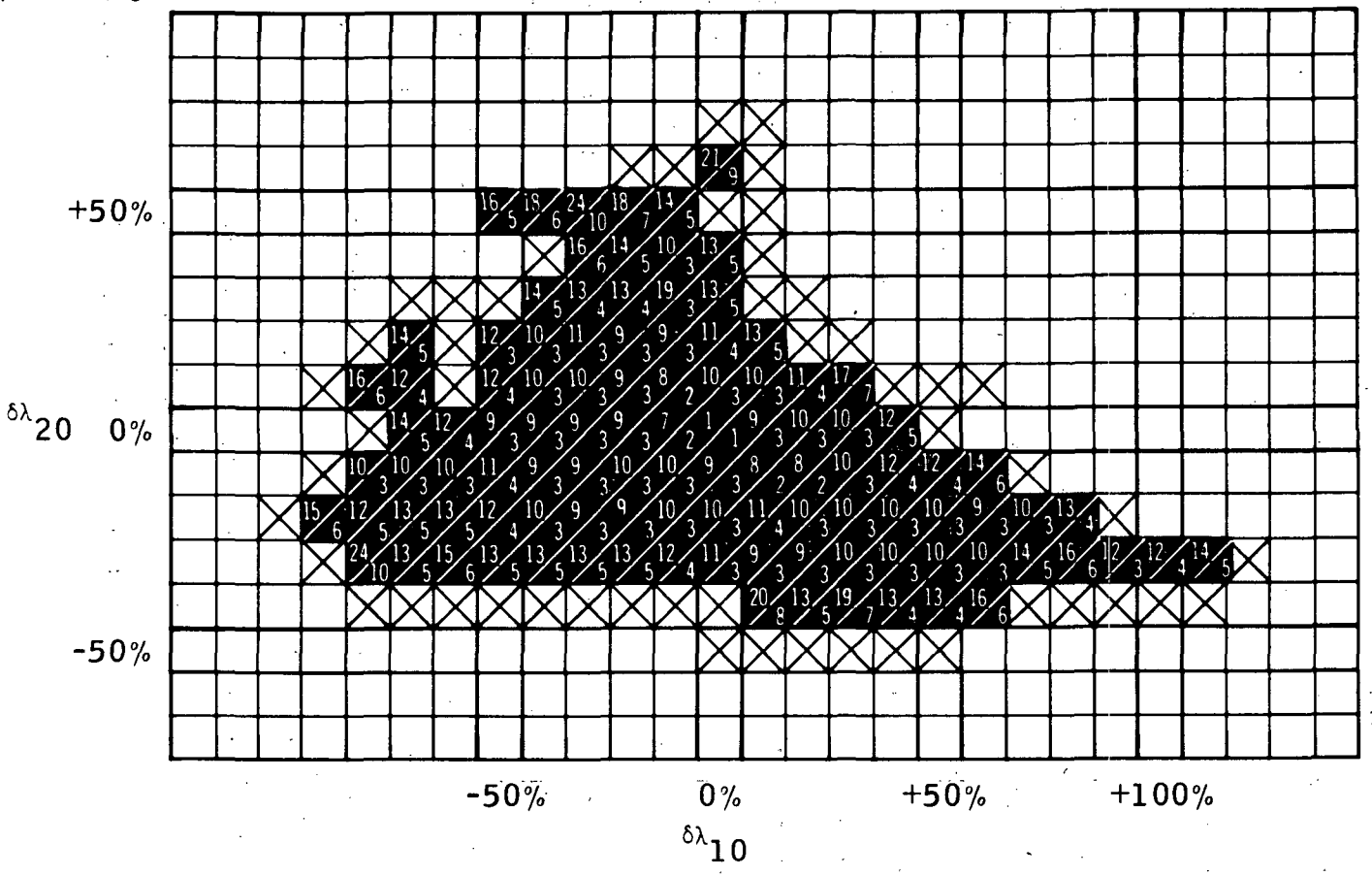


Figure 12.- Convergence envelope for the MAF using iteration scheme 2, initial iteration factor of 50% terminal time error of 0% and update integer of 6.

increasing the updating integer. The total number of iterations required increased, but the number of adjoint integrations decrease as the updating integer is increased. This trend continues until the updating integer reaches four or six and this appears to be a point of diminishing return for this particular problem.

It becomes apparent that the initial value of the iteration factor has a pronounced effect on the convergence envelope size, and in most cases convergence time as well. An initial value of iteration factor of 20 percent, with either iteration scheme, produces a significantly larger envelope than the ones for 50 percent shown in Figures 7 through 12. This increase in envelope size is accompanied by a significant increase in the required computer convergence time for Iteration Scheme 1. Figure 13 illustrates this influence of the initial values of iteration factor on the convergence time for the particular but representative cases where the Lagrange multiplier and terminal time errors are as indicated on the figure. For Case 1, where the two Lagrange multipliers and terminal time errors are -10, -10, and 20 percent, respectively, the largest values of initial iteration factor result in the most favorable convergence times. On the other hand, for the case where the initial error is larger, as illustrated by Case 2 where the Lagrange multipliers and terminal time errors are -20, 10, and 20 percent, respectively, some intermediate value of initial iteration factor results in the most favorable time.

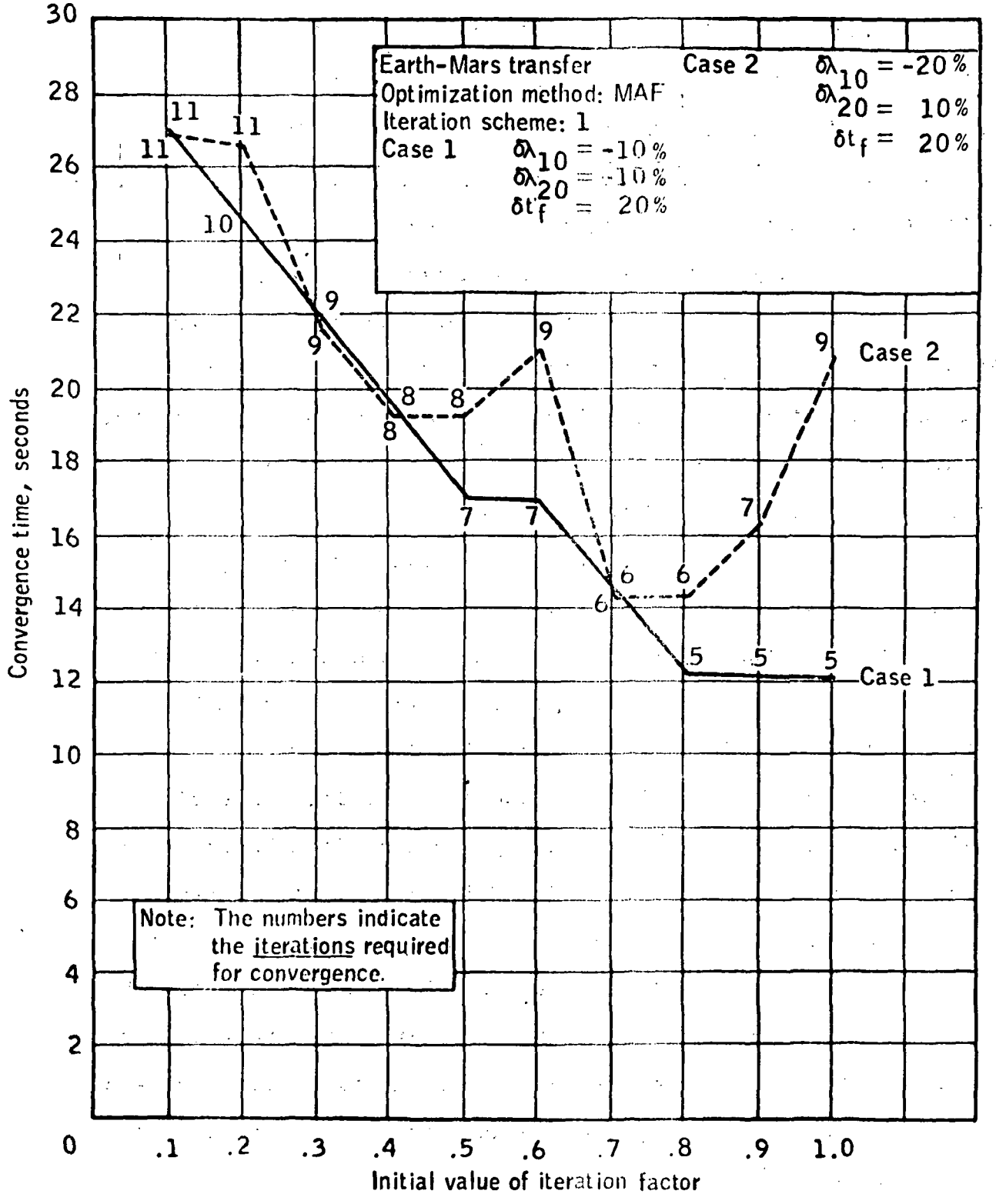


Figure 13. - Convergence time as a function of the initial value of iteration factor for the MAF using iteration scheme 1.

Figure 13 also reveals the existence of an uncertainty about the selection of the initial iteration factor. When a problem is first attacked, one has little or no feel for the percentage correction to request. A low initial value for the iteration factor is usually selected because it is expected that this results in a large envelope of convergence. A low initial iteration factor results in a convergence time penalty as shown in Figure 13. However, in some situations a high value for the initial iteration factor results in a convergence time penalty. It is not known how to determine the best initial iteration factor before a series of investigations is made.

Iteration Scheme 2 attempts to overcome this problem by seeking the largest iteration factor that can be used, without a trajectory divergence, before the time consuming adjoint integration is made. Since only forward integrations are made in bringing the iteration factor from a low initial value to the best value, the time penalty is reduced. The influence of initial iteration factor on the convergence time is illustrated in Figure 14 for Iteration Scheme 2. This plot may be compared to one of the cases in Figure 13, and it is easily seen that for low initial values of the iteration factor the time penalty is not so severe. The objection to an initial low iteration factor is removed now, and yet good convergence possibilities remain because large envelopes of convergence are associated with low initial iteration factors.

The influence of the update integer on convergence times is illustrated in Figures 15, 16, 17, and 18. These envelopes

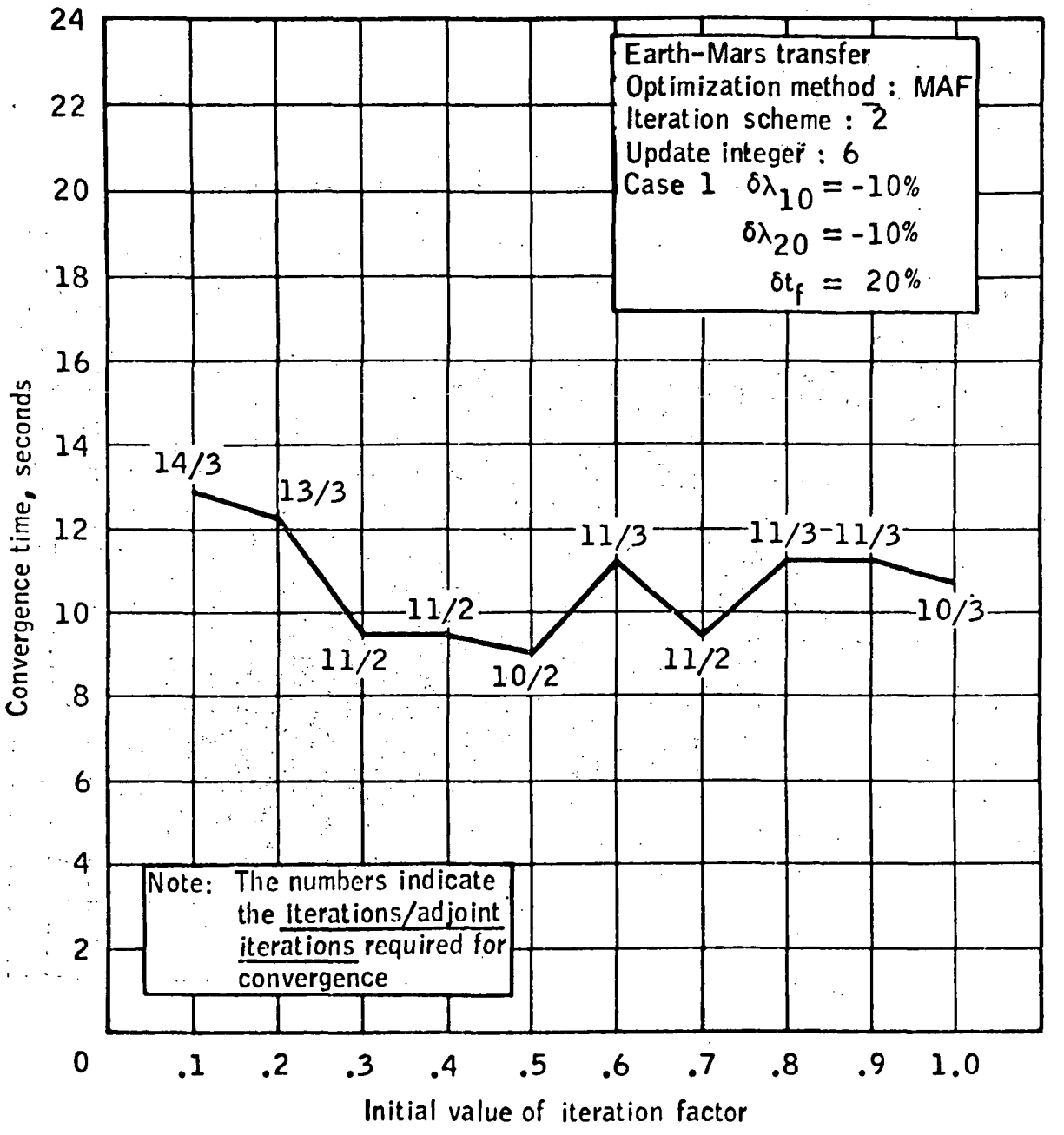
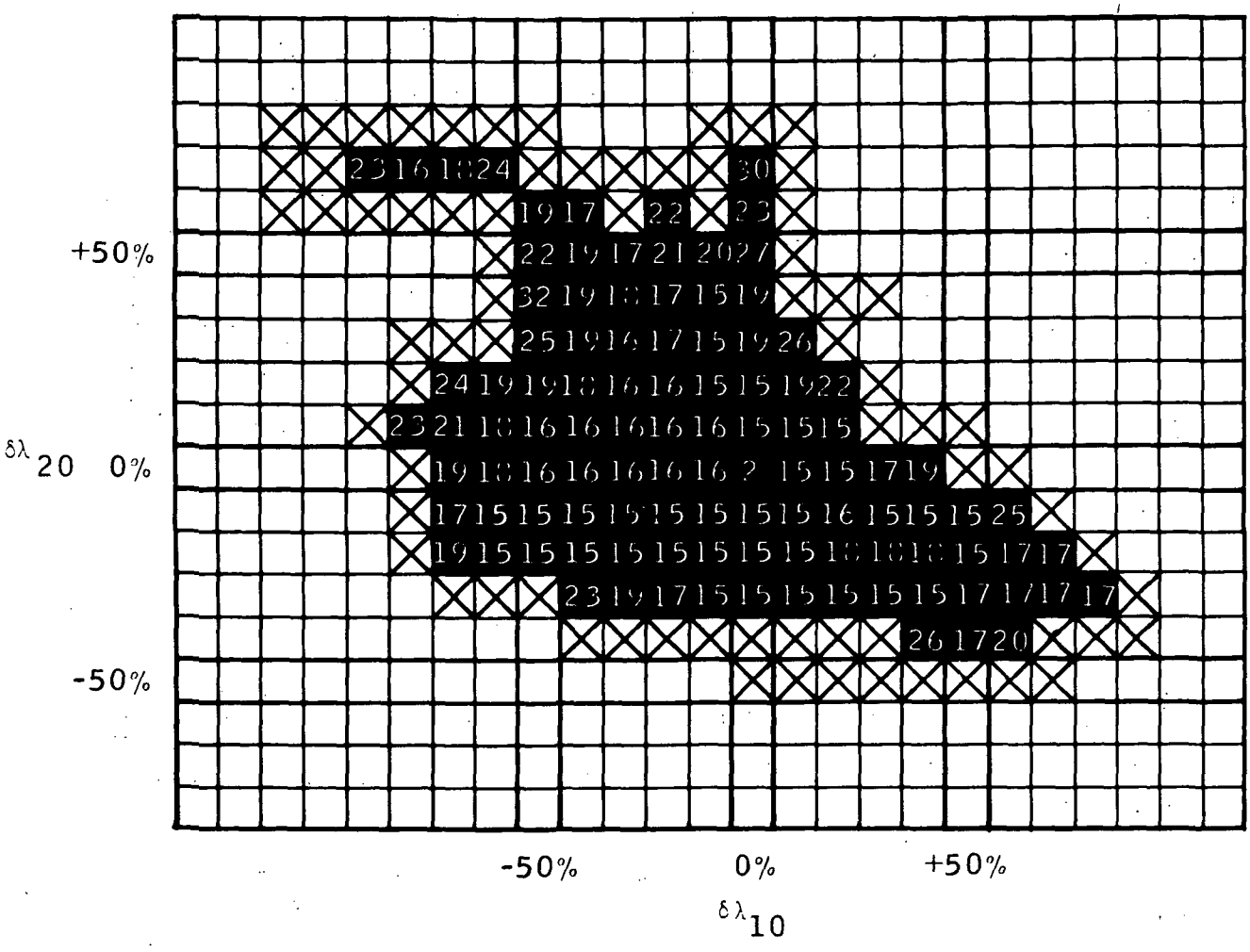


Figure 14. - Convergence time as a function of the initial value of iteration factor for the MAF using iteration scheme 2.

Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 1 and 2
Initial iteration factor: 50%
Terminal time error: 0%
Update integer: 1



Note: *The numbers indicate the time in seconds required for convergence*

Figure 15.- Convergence envelope for the MAF using iteration schemes 1 and 2, initial iteration factor of 50%, terminal time error of 0% and update integer of 1.

Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 2
Initial iteration factor: 50%
Terminal time error: 0%
Update integer: 2

Note: The numbers indicate the time in seconds required for convergence

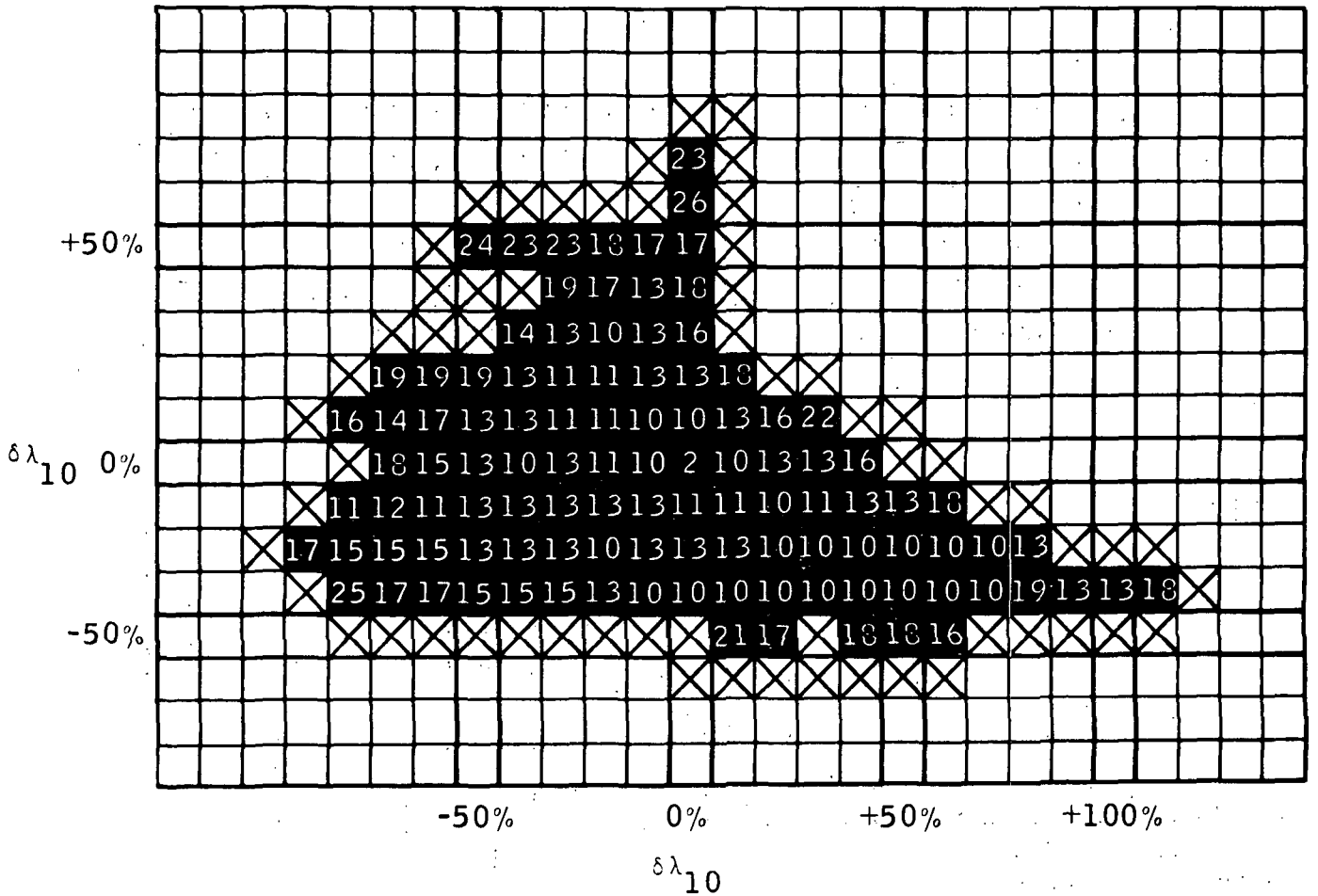


Figure 16.- Convergence envelope for the MAF using iteration scheme 2, initial iteration factor of 50%, terminal time error of 0% and update integer of 2.

Earth-Mars transfer
Optimization method: MAF
Iteration scheme: 2
Initial iteration factor: 50%
Terminal time error: 0%
Update integer: 4

Note: The numbers indicate the time in seconds required for convergence

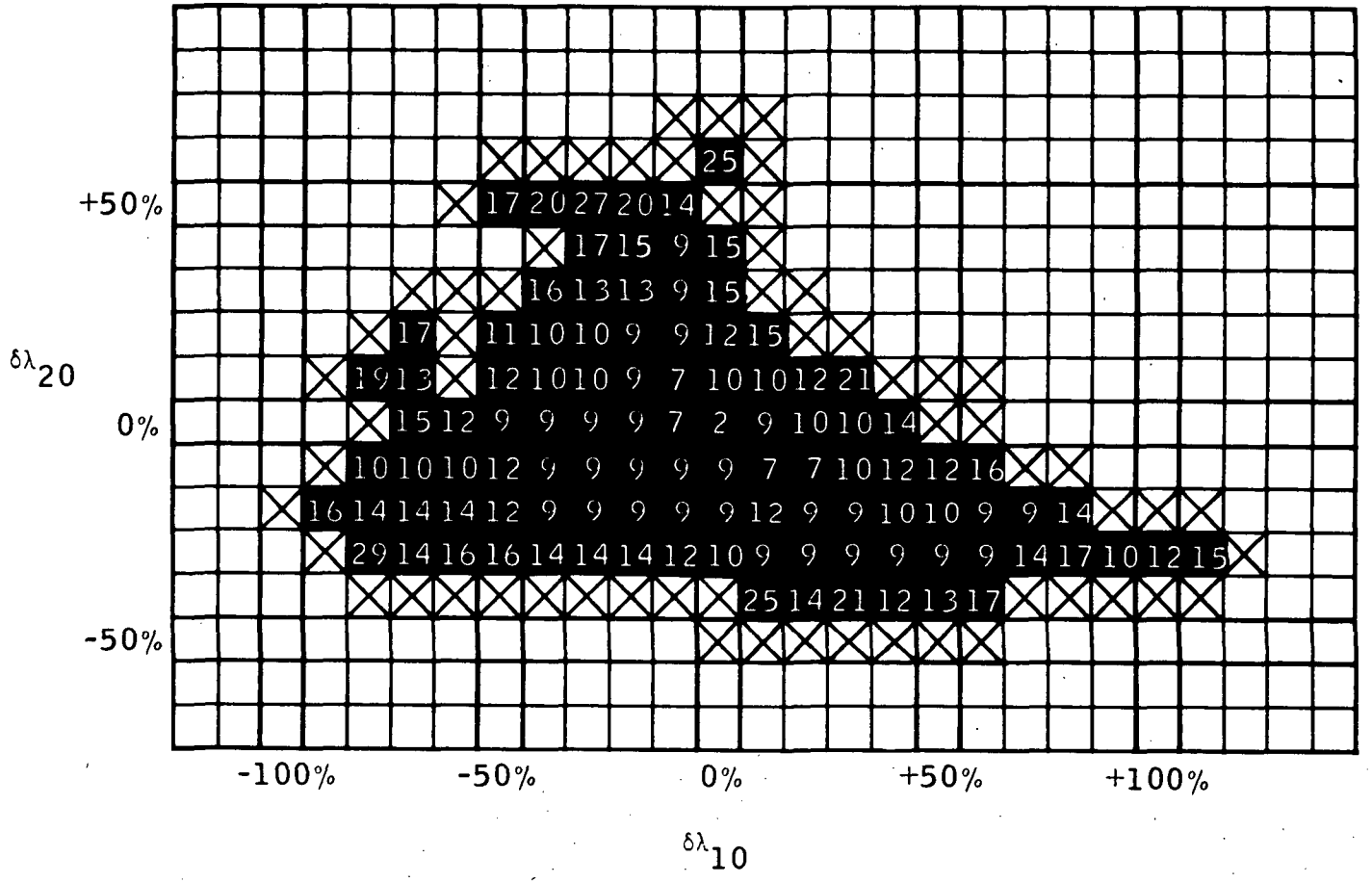


Figure 17.- Convergence envelope for the MAF using iteration scheme 2, initial iteration factor of 50%, terminal time error of 0% and update integer of 4.

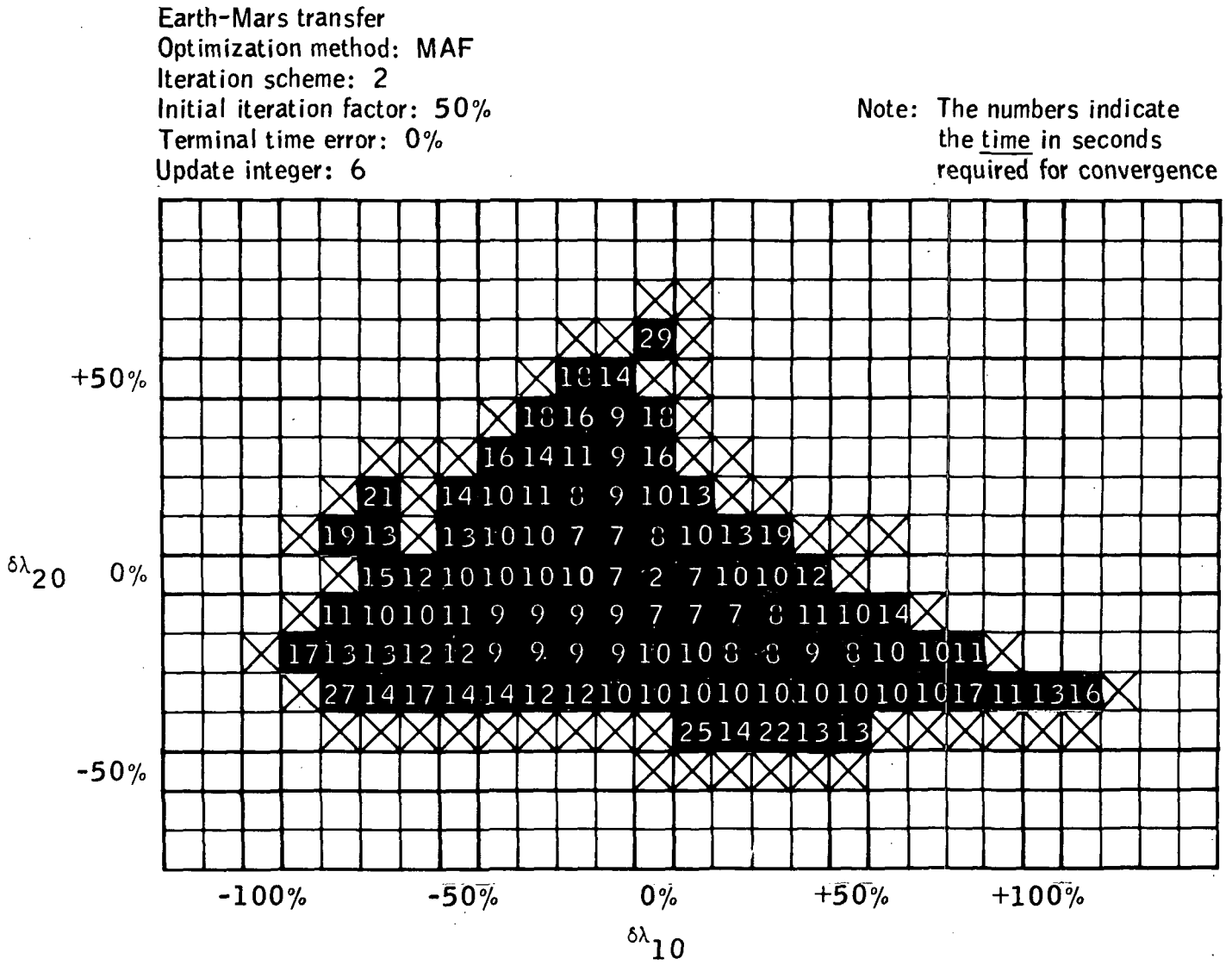


Figure 18.- Convergence envelope for the MAF using iteration scheme 2, initial iteration factor of 50%, terminal time error of 0% and update integer of 6.

correspond to the envelopes in Figures 8, 10, 11, and 12, but indicate the convergence times rather than the required iterations. A most interesting characteristic of Iteration Scheme 2 is revealed. For a given initial iteration factor of 50 percent, the convergence times are generally reduced by increasing the updating integer to the four to six range. Larger values of the updating integer result in higher convergence times. It is expected that for this problem the best update integer approximately equals the number of steps required between the initial value of the iteration factor and unity.

It is very interesting to take a specific and representative example, and examine the norm of the terminal constraints as a function of computation time. Figure 19 shows the terminal dissatisfaction norm decreasing for Iteration Scheme 1 for initial values of the iteration factor of 20, 50, 70, and 100 percent. Not only is the increase in convergence time for the smaller iteration factors evident, but the characteristics of the convergence rate are also seen. Figure 20 illustrates these same characteristics for Iteration Scheme 2 using an initial iteration factor of 50 percent. The norm of the terminal dissatisfaction is plotted as a function of computation time for update integers of 1, 2, 4, and 6. With an update integer of six, the convergence time is approximately reduced by 50 percent when compared to the extreme case where the integer is unity.

In an effort to determine some of the complications associated with solving a different problem, the atmospheric

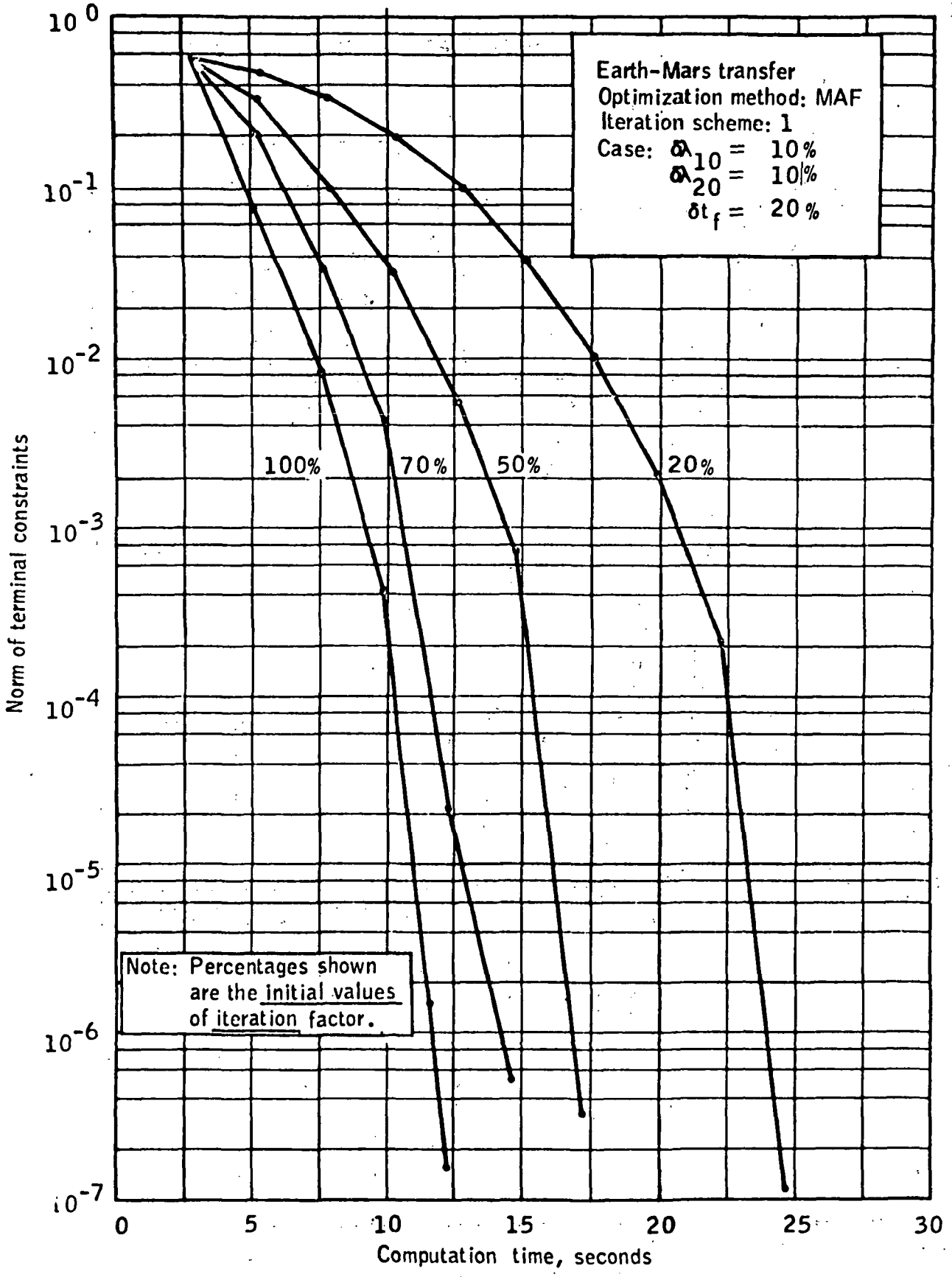


Figure 19. - Norm of terminal constraints as a function of computation time for the MAF using iteration scheme 1.

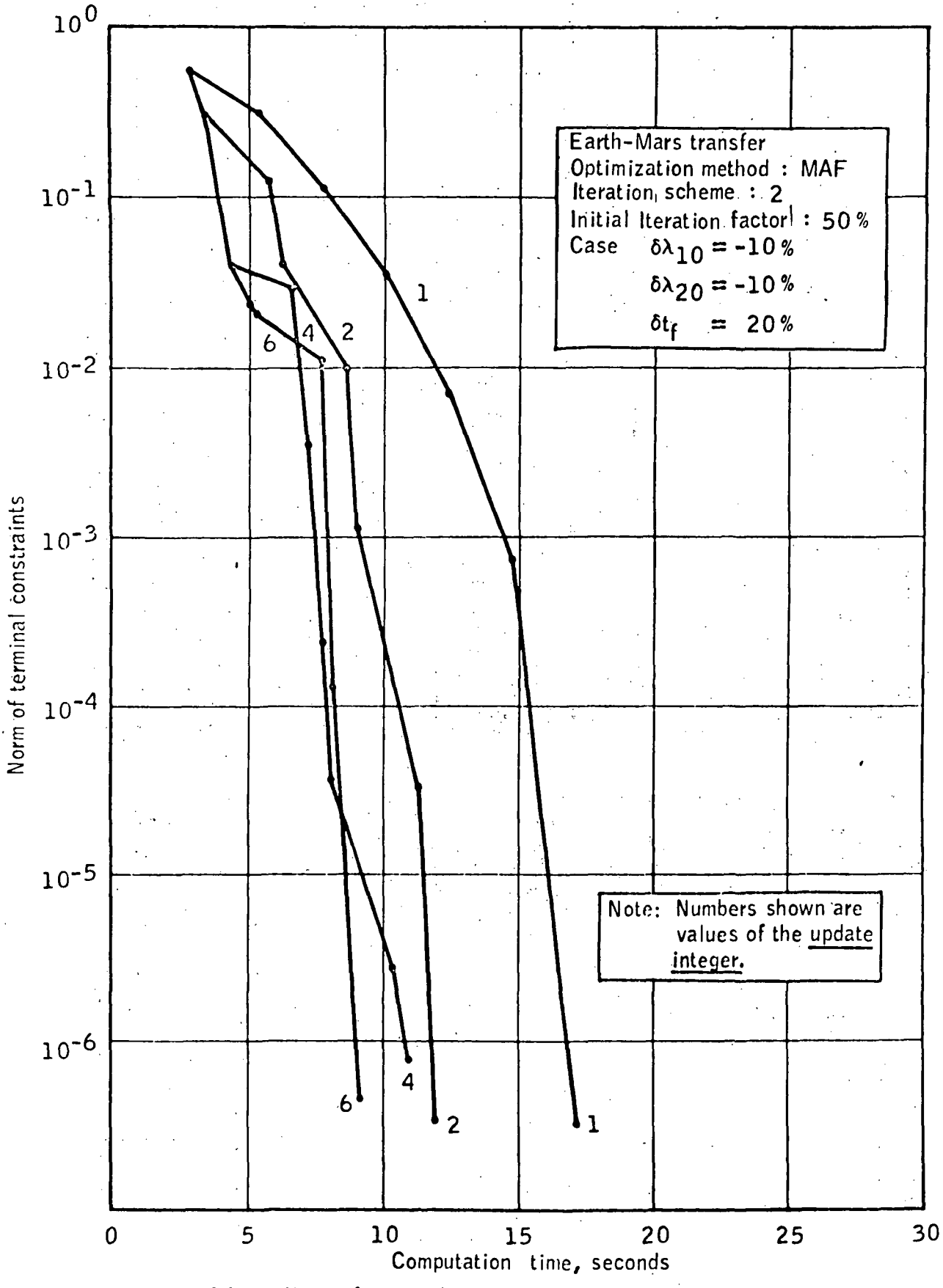


Figure 20. - Norm of terminal constraints as a function of computation time for the MAF using iteration scheme 2.

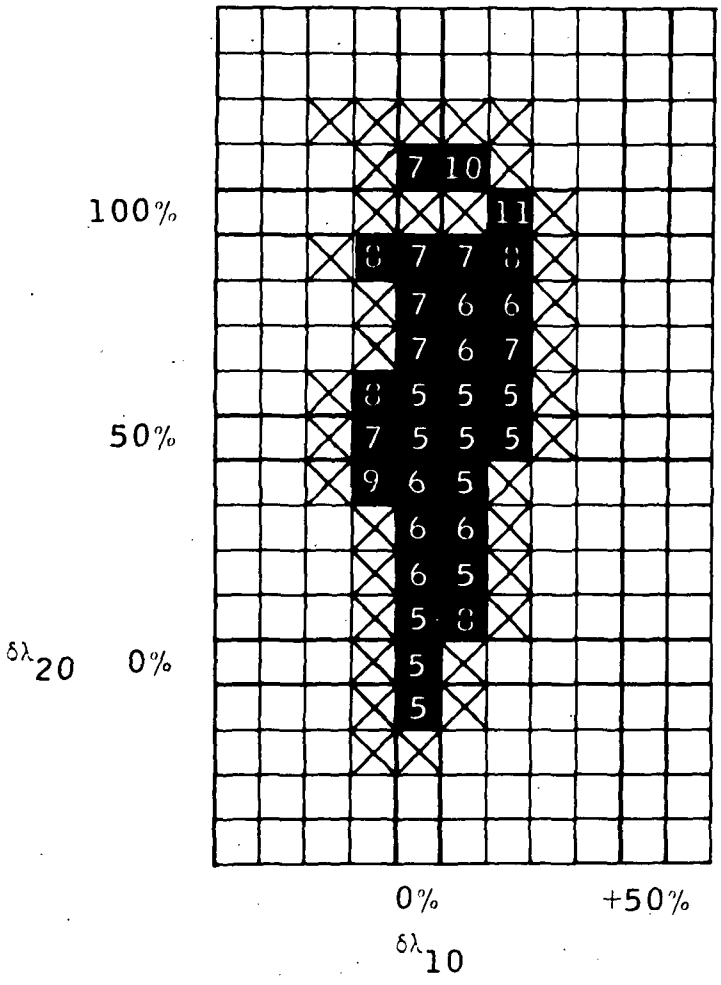
Earth launch to circular orbit described in Appendix A.2 was formulated and solved. These results are shown in Figures 21, 22, and 23. It was discovered, for the Earth launch problem, that the convergence envelopes were less sensitive to terminal time errors than for the Earth-Mars transfer. Hence, the plots shown are the same as for previous cases with the exception that terminal time variations are only 10 percent.

It is obvious from the figures that the method is relatively sensitive to λ_{10} errors and relatively insensitive to λ_{20} errors. This Earth launch example reveals some of the same characteristics seen for the Earth-Mars transfer, namely, as the terminal time error increases the convergence envelope increases in size and moves downward. This downward movement means a reduction of negative λ_2 error sensitivity.

One interesting characteristic, not seen in the Earth-Mars transfer example, is that when the λ_{20} error is 100 percent, considerable convergence difficulty is experienced. This case corresponds to the initial control angle of 90 degrees. It is rather remarkable that convergence still results for some cases where the initial control angle is greater than 90 degrees.

In summary, for Iteration Scheme 1 the envelope of convergence increases with positive increases in terminal time error, for a given initial iteration factor. The envelope size is increased further with a reduction of initial iteration factor, but unfortunately the convergence time is increased. The convergence envelope for Iteration Scheme 2 is also increased by

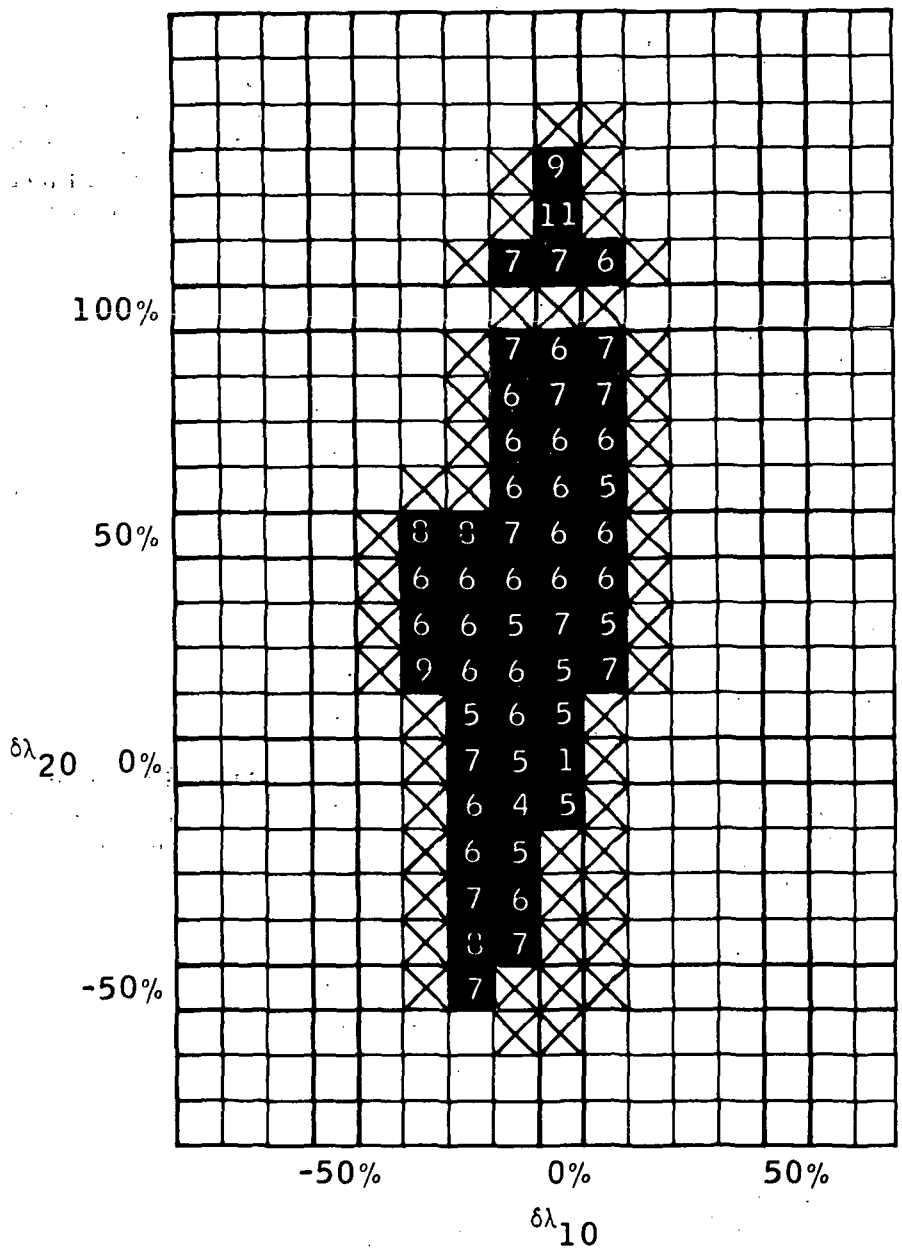
Earth launch
Optimizationm
Earth launch
Optimization method: MAF
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: -10%



Note: The numbers indicate the iterations required for convergence

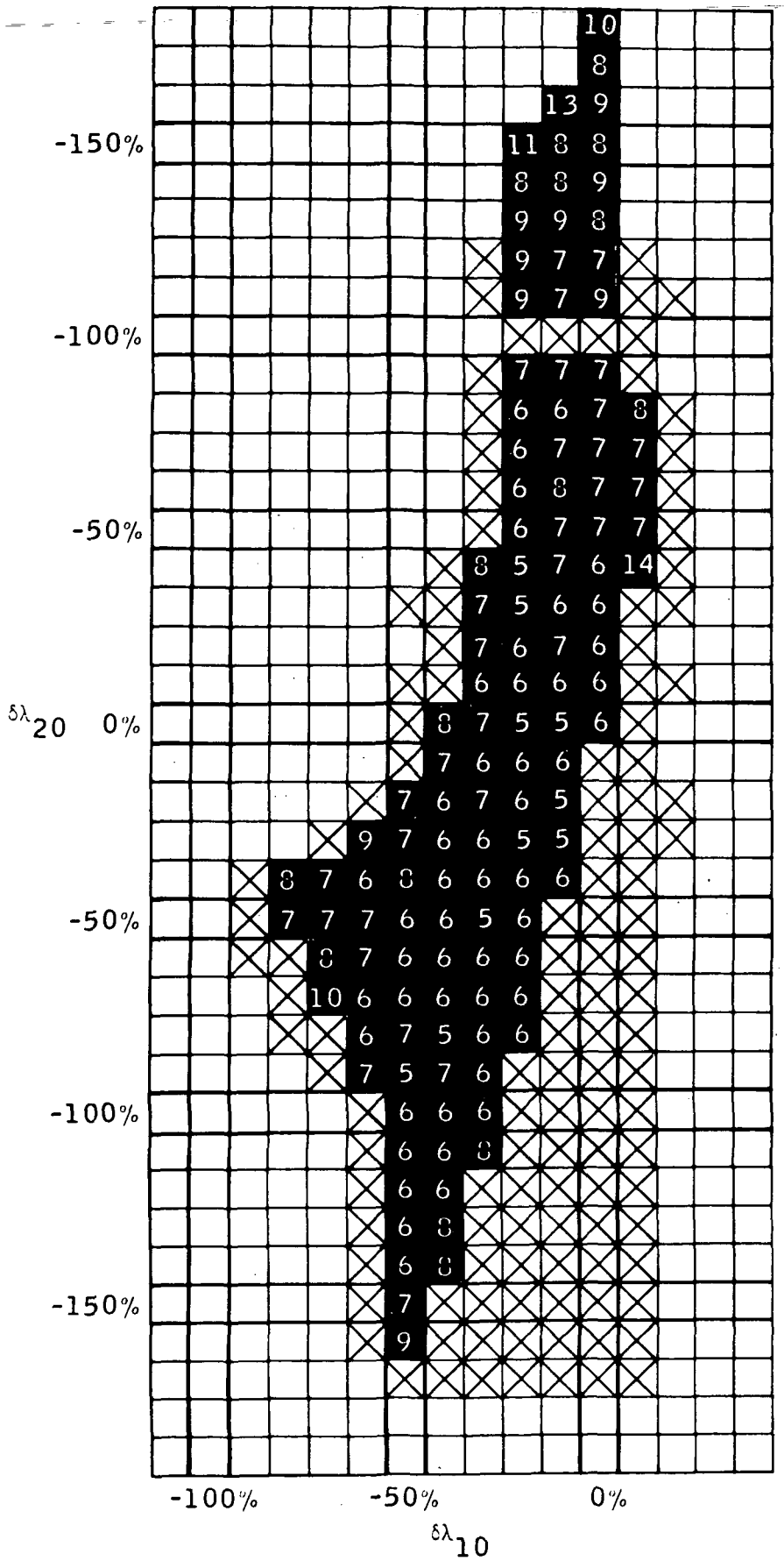
Figure 21.- Convergence envelope for the MAF using the normal iteration scheme, initial iteration factor of 100% and terminal time error of -10% (Earth launch).

Earth launch
Optimization method: MAF
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: 0%



Note: The numbers indicate the iterations required for convergence

Figure 22.- Convergence envelope for the MAF using the normal iteration scheme, initial iteration factor of 100% and terminal time error of 0% (Earth launch).



Earth launch
Optimization method: MAF
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: 10%

Note: The numbers indicate the iterations required for convergence

Figure 23.- Convergence envelope for the MAF using the normal iteration scheme, initial iteration factor of 100% and terminal time error of 10% (Earth launch).

reducing the initial iteration factor, for a given update integer. For a given initial iteration factor, the convergence time is reduced by increasing the update integer. The best times result for update integers of approximately six, and increased times result for further increases in the integer.

The significant fact is that Iteration Scheme 2 is superior to Iteration Scheme 1 because low, and hence safe, initial values of the iteration factor may be used without resulting in an unreasonably large convergence time.

The application of this optimization method to a different problem resulted in approximately the same general convergence characteristics.

6.3.2 Method of Perturbation Functions

The required formulation as discussed in Section 3.2 is simple and straightforward, and even more natural than MAF since the perturbation equations are used directly. A general discussion of the applications is presented in Appendix A.2 and a specific application of the MPF is made in Appendix A.2.2.

The programming effort requires the forward integration of the eight differential equations of motion and the Euler differential equations. The eight perturbation equations must also be integrated forward, and this must be done with three different starting vectors. The coefficients for these perturbation equations may be formed as needed and no storage is required. This represents a decided advantage over the MAF,

especially when the problem is of large dimension, because the back spacing of tapes is not necessary. The programming complexity is reduced also because no checks are required for the acquisition of proper coefficients, i.e., the coefficients are simply formed as the forward integration is made. It may also be noted that one less integration is required for the MPF as opposed to the MAF, and this results in less total integration time.

The integration of the perturbation equations requires a large percentage of the total computational time. It is conceivable that the same numerical accuracy might result when a variable integration step size is used, however, this increases the programming complexity considerably. A constant step size was selected for the integration of all equations.

The computer program that uses the MPF requires two initially assumed Lagrange multipliers and an assumed terminal time. These estimates require a familiarity with the physical problem and, to some degree, experience. The computer program is built such that only the subroutines containing the differential equations of motion, the Euler-Lagrange equations, and the perturbation equations must be changed to solve different problems, and the effort is comparable to that required for the MAF.

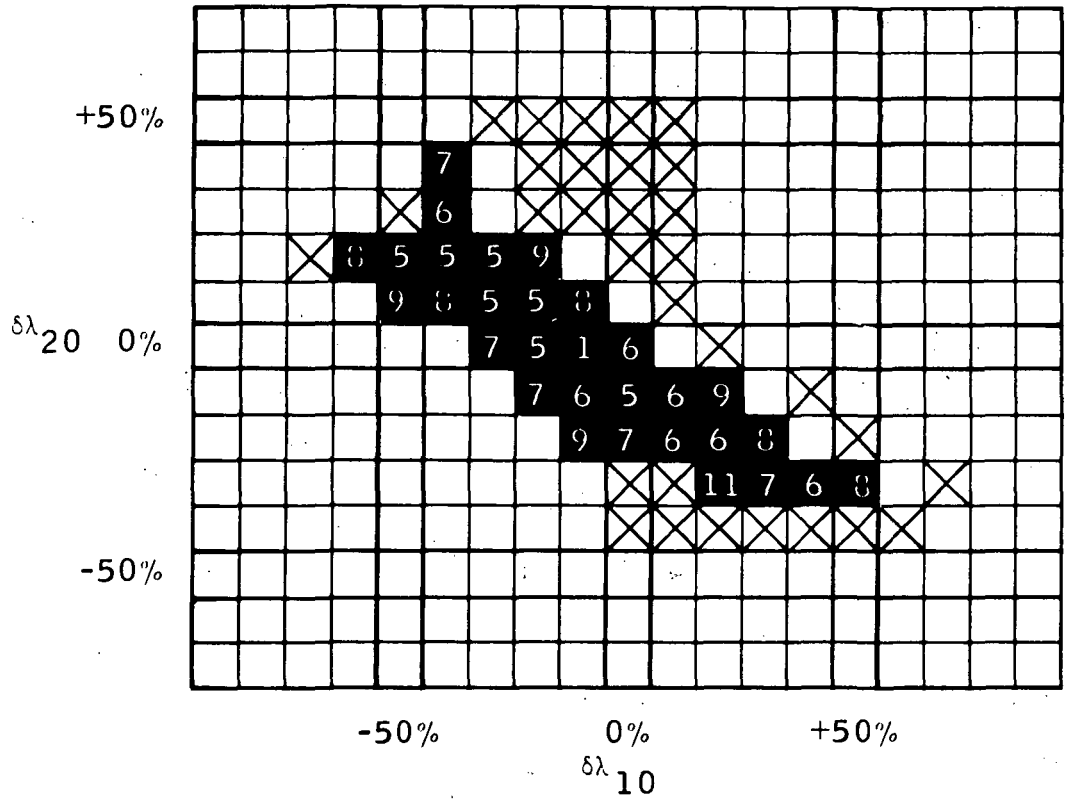
Iteration Scheme 1 requires very little computer logic in addition to the Normal Scheme of requesting 100 percent terminal constraint satisfaction on each iteration. Operation is simply transferred to a subroutine where the iteration factor is

altered in accordance with the terminal norm criteria explained in Section 3.3. The process is essentially the same as that for the MAF.

Iteration Scheme 2 requires some additional programing and storage, and is comparable to that required for the MAF. Basically, the scheme is such that the iteration factor is increased, omitting a perturbation integration, until either the norm of the terminal constraints diverges or a specified number of nominals have been generated. If the norm diverges, the last convergent trajectory is used as a nominal, and hence this trajectory must be saved until it is determined whether or not it will be needed. The storage problem can be eliminated by simply regenerating the last convergent trajectory.

An extensive analysis of the MPF is not made since the theoretical development in Section 3.2 shows that exactly the same algebraic equation used for the MAF is used to determine the corrections. The only difference between the MAF and MPF is that one less integration is required for MPF, and therefore a reduced convergence time is expected. The envelopes of convergence for Iteration Scheme 1 using initial iteration factors of 100 and 50 percent, respectively, are shown in Figures 24 and 25. The obvious fact is that the envelopes have the same size and shape as the corresponding envelopes for the MAF shown in Figures 5 and 8, and the numbers on the figures indicate an equal number of iterations are required. Figures 26 and 27 illustrate the convergence times for the above cases. A

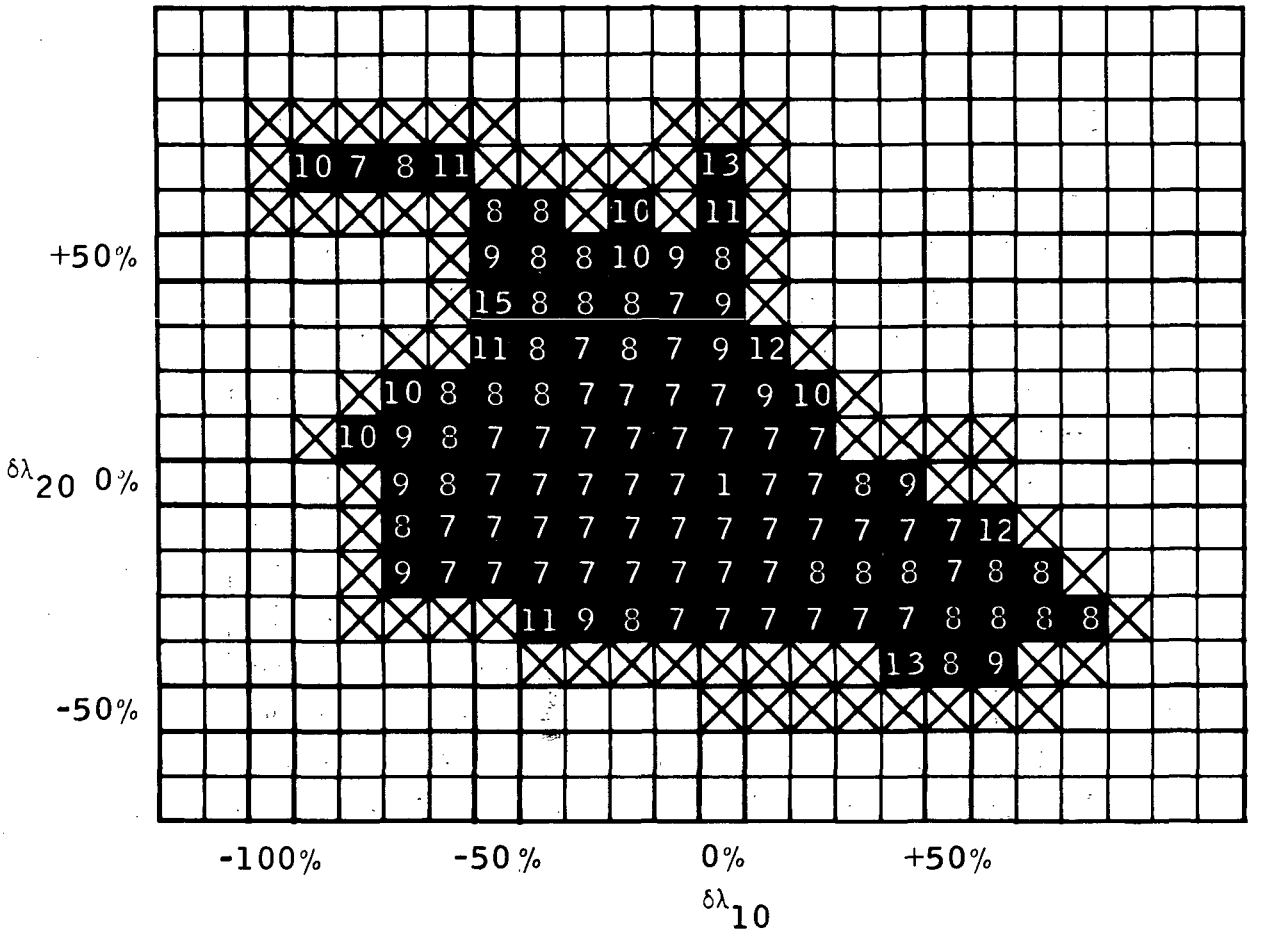
Earth-Mars transfer
Optimization method: MPF
Iteration scheme: 1
Initial iteration factor: 100%
Terminal time error: 0%



Note: The numbers indicate the iterations required for convergence

Figure 24.- Convergence envelope for the MPF using iteration scheme 1, initial iteration factor of 100% and terminal time error of 0%.

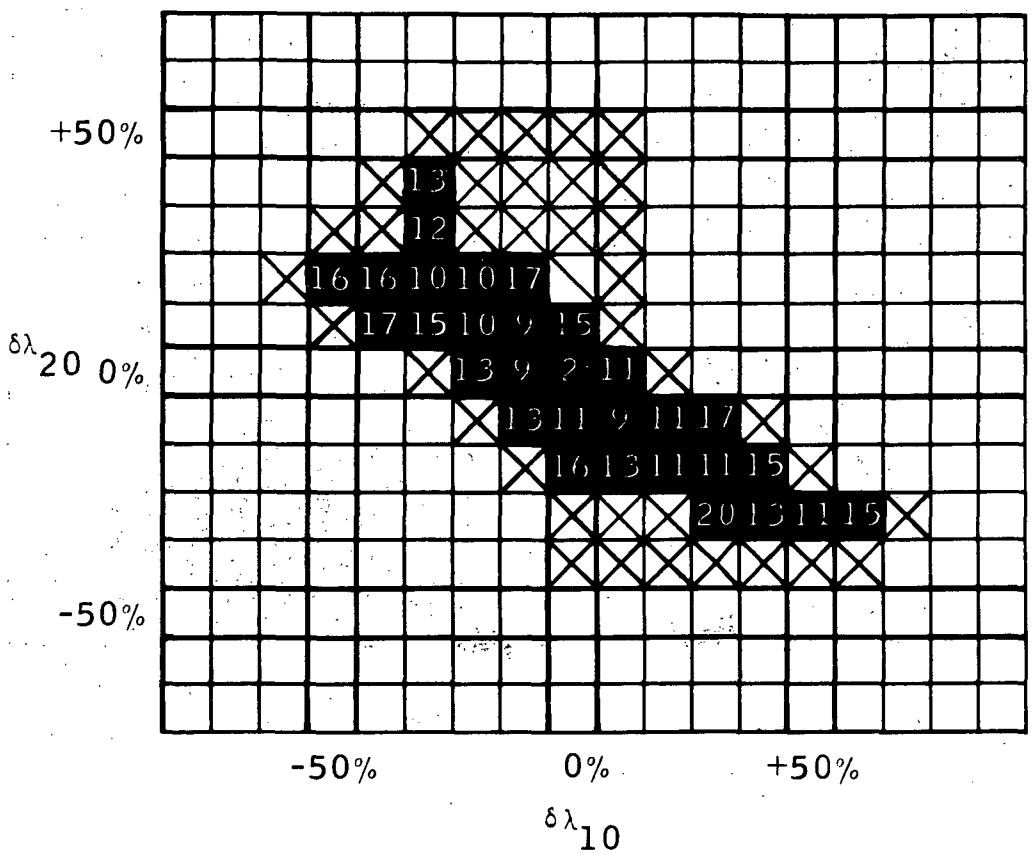
Earth-Mars transfer
Optimization method: MPF
Iteration scheme: 1
Initial iteration factor: 50%
Terminal time error: 0%



Note: The numbers indicate the iterations required for convergence

Figure 25.- Convergence envelope for the MPF using iteration scheme 1, initial iteration factor of 50% and terminal time error of 0%.

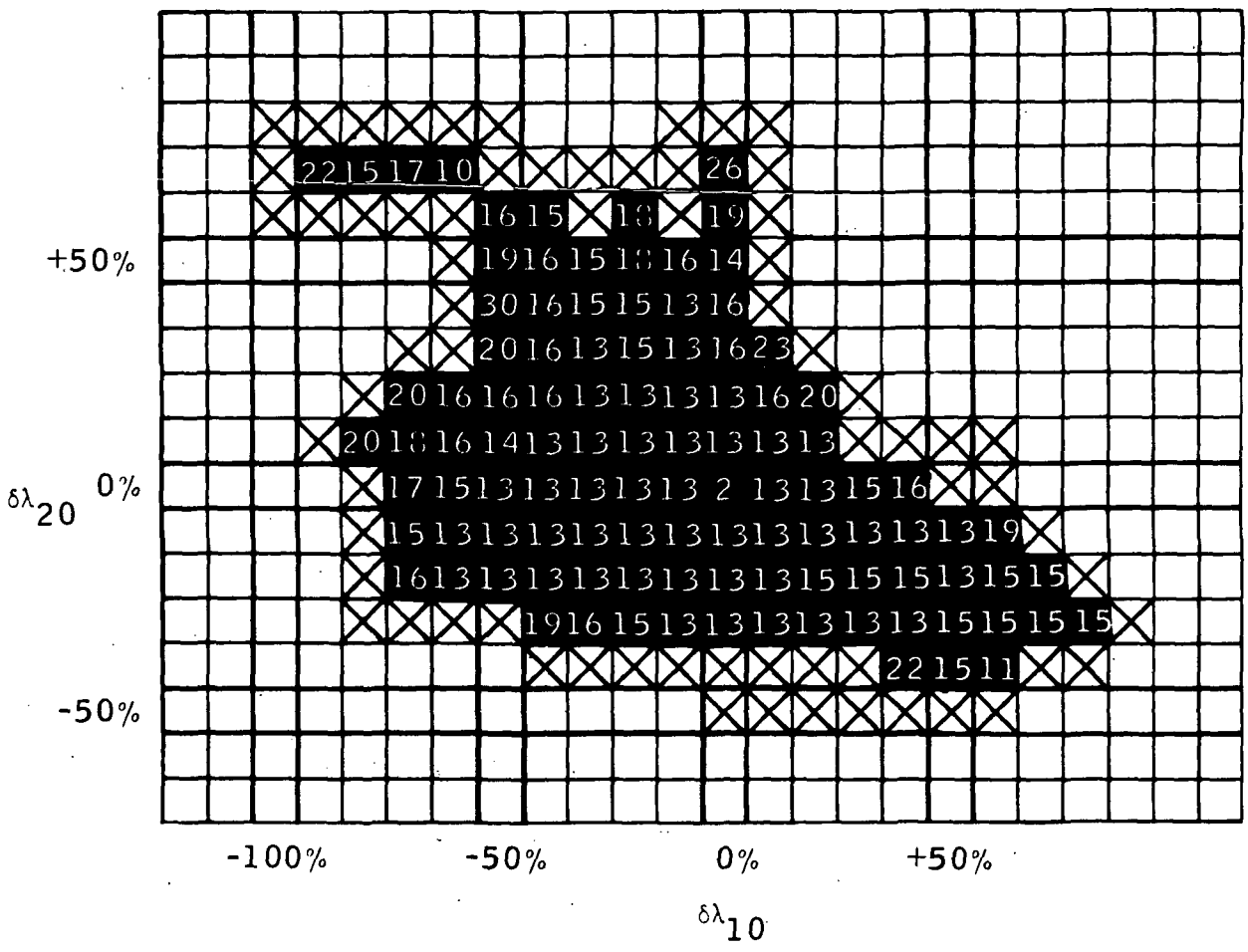
Earth-Mars transfer
Optimization method: MPF
Iteration scheme: 1
Initial iteration factor: 100%
Terminal time error: 0%



Note: The numbers indicate the time in seconds required for convergence

Figure 26.- Convergence envelope for the MPF using iteration scheme 1, initial iteration factor of 100% and terminal time error of 0%.

Earth-Mars transfer
Optimization method: MPF
Iteration scheme: 1
Initial iteration factor: 50%
Terminal time error: 0%



Note: The numbers indicate the time in seconds required for convergence

Figure 27.- Convergence envelope for the MPF using iteration scheme 1, initial iteration factor of 50% and terminal time error of 0%.

comparison of the convergence times may be made between the MAF and MPF by comparing the times shown in Figures 15 and 27, respectively. It is seen that the MAF must integrate a comparable set of differential equations four times rather than only three, as required by the MPF. Iteration Scheme 2 for the MPF was not programmed.

The significant fact is that the MPF results in the same envelope of convergence and requires the same number of iterations as the MAF, but approximately 20 percent less computer time is required because one less integration is needed.

6.4 Quasilinearization Methods

The comparison and discussion of the Quasilinearization Methods will consist of two separate analyses. The Method of Generalized Newton-Raphson, including the normal procedure and Iteration Scheme 1 is discussed first. The Modified Quasilinearization Method including the normal procedure and Iteration Scheme 2 is discussed last. The Modified Method of Generalized Newton-Raphson is also discussed briefly, but the MQM is emphasized. The discussion content will include the applicable items listed in the Section 6.2.

6.4.1 Method of Generalized Newton-Raphson

The required formulation of the Method of Generalized Newton-Raphson as discussed in Section 4.1 is simple and relatively easy to apply, although this particular method is not

capable of handling terminal constraint functions or determining the terminal time in an efficient manner. For these reasons, an extensive investigation of this method is not made. However, several runs are made, and spot comparisons illustrate its effectiveness with respect to the other methods.

The programming effort requires the forward integration of the homogeneous parts of eight linearized differential equations of motion and the Euler differential equations. Also the nonhomogeneous parts are integrated forward once, and all coefficients for the solution of a linear system must be included for use after each trajectory iteration. When convergence is obtained for the specified value of terminal time, a time iteration is made by making a scalar application of the Newton-Raphson technique.

If the solutions to both the homogeneous and nonhomogeneous equations are stored, a new nominal is immediately available. However, to conserve storage only the terminal values of the solutions are stored and the next nominal is simply generated by an additional integration.

The current trajectory is generated from the preceding trajectory, however, after a positive correction of terminal time has been made, no previous information is available. This fact represents a problem that does not exist for the MAF or MPF. The program is written so that a linear extension of all the variables of the previous nominal is made to provide information for

the current trajectory.

The computer program that uses the MGNR requires two initially assumed Lagrange multipliers, an assumed terminal time, and an initial trial solution consisting of the time histories of all eight variables. The estimates require a familiarity with the physical problem to insure that the assumed quantities are close enough to optimal that convergence will result. The significant difference between MGNR and MAF or MPF is that a complete solution must be assumed rather than just initial starting values of the variables. If no reasonable solution can be decided upon, the nonlinear equations may be integrated to provide the first solution. However, in the more complex problems, this solution may not be adequate to result in convergence.

The program is built such that only the subroutines containing the nonhomogeneous and homogeneous equations and the trial solution must be changed to solve different problems. A constant integration step size was selected for all integrations.

The Normal Scheme of the MGNR is that of making trajectory iterations, requesting 100 percent correction in the terminal constraints, until convergence results for the assumed terminal time. Then a time iteration is made and the process continued. Iteration Scheme 1 requires very little additional computer logic. This scheme amounts to avoiding time iterations until the present metric becomes less than the previous metric. The logic is simply inserted in the program, and an additional subroutine is not used.

A typical example of the convergence characteristics of the MGNR is shown in Figure 28. This illustration shows how the metric decreases as a function of computation time for the case where the Lagrange multipliers and terminal time errors are -10, -10, and 20 percent, respectively. A linear initial trial solution is used and this solution is represented by long dashed lines in Figure A.2.1. Trajectory iterations are made until the metric is less than 10^{-5}, then a time iteration is made. During the initial stages, the time iteration essentially destroys the reduced metric that has just been obtained. This characteristic is not quite so severe when terminal time errors are small.

The convergence characteristics for the same example, using Iteration Scheme 1 are shown in Figure 29, and a significant reduction in computation time is evident. This scheme appears superior to the normal procedure, but it must be pointed out that a theoretical analysis of this scheme has not been made to define a bounds for convergence. For a given terminal time, the convergence proof given by McGill (14) applies, but the time iterations could be so poor that divergence would result. The examples in Figures 28 and 29 show that the Iteration Scheme 1 results in a convergence time that is 43 percent less than that required by the Normal Scheme.

The Modified Method of Generalized Newton-Raphson, discussed in Section 4.1, is modified in the sense that a change in the independent variable is made to eliminate the cumbersome determination of terminal time. One advantage of this method is

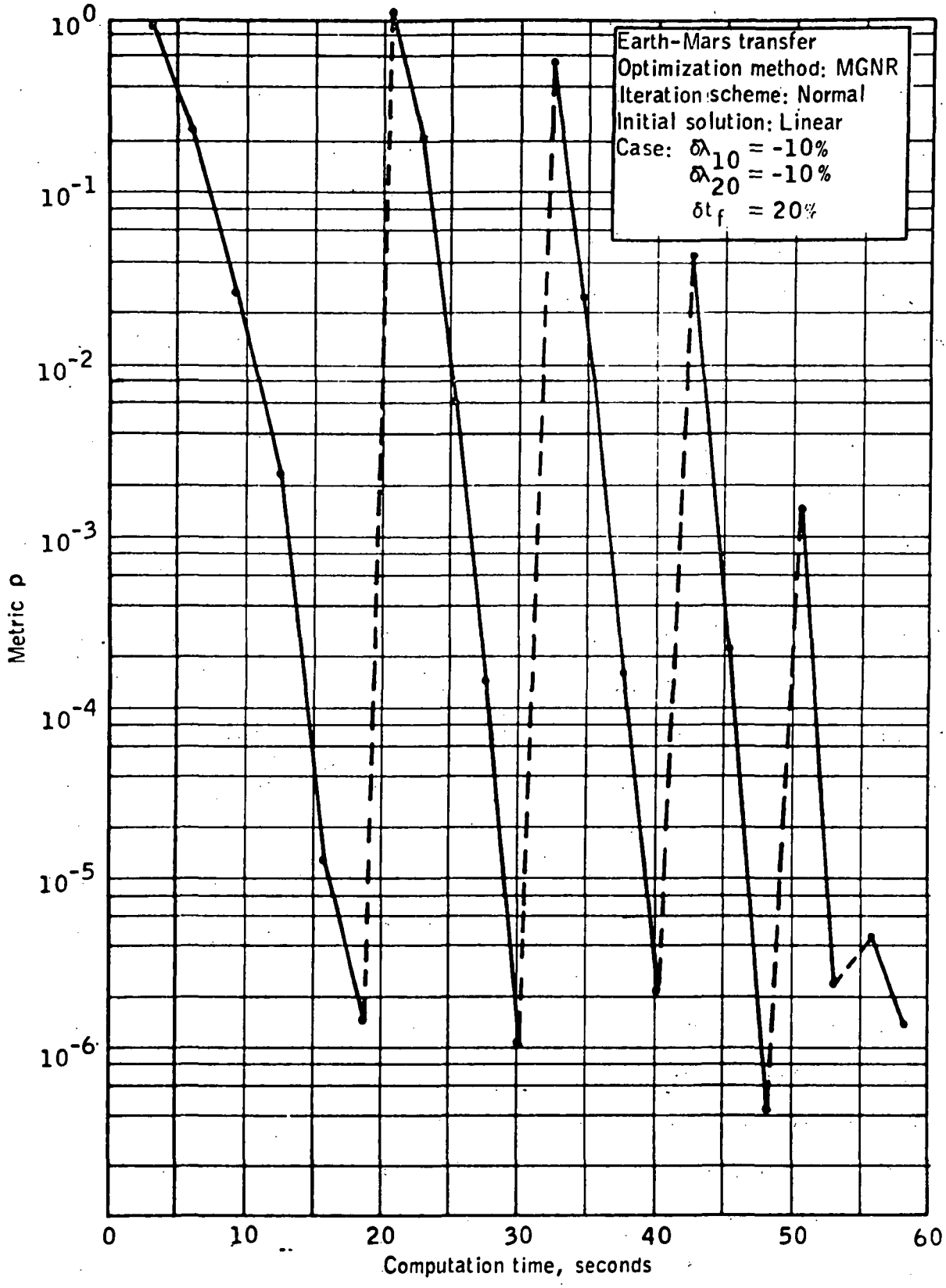


Figure 28. - Metric ρ as a function of computation time for the MGNR using the normal iteration scheme.

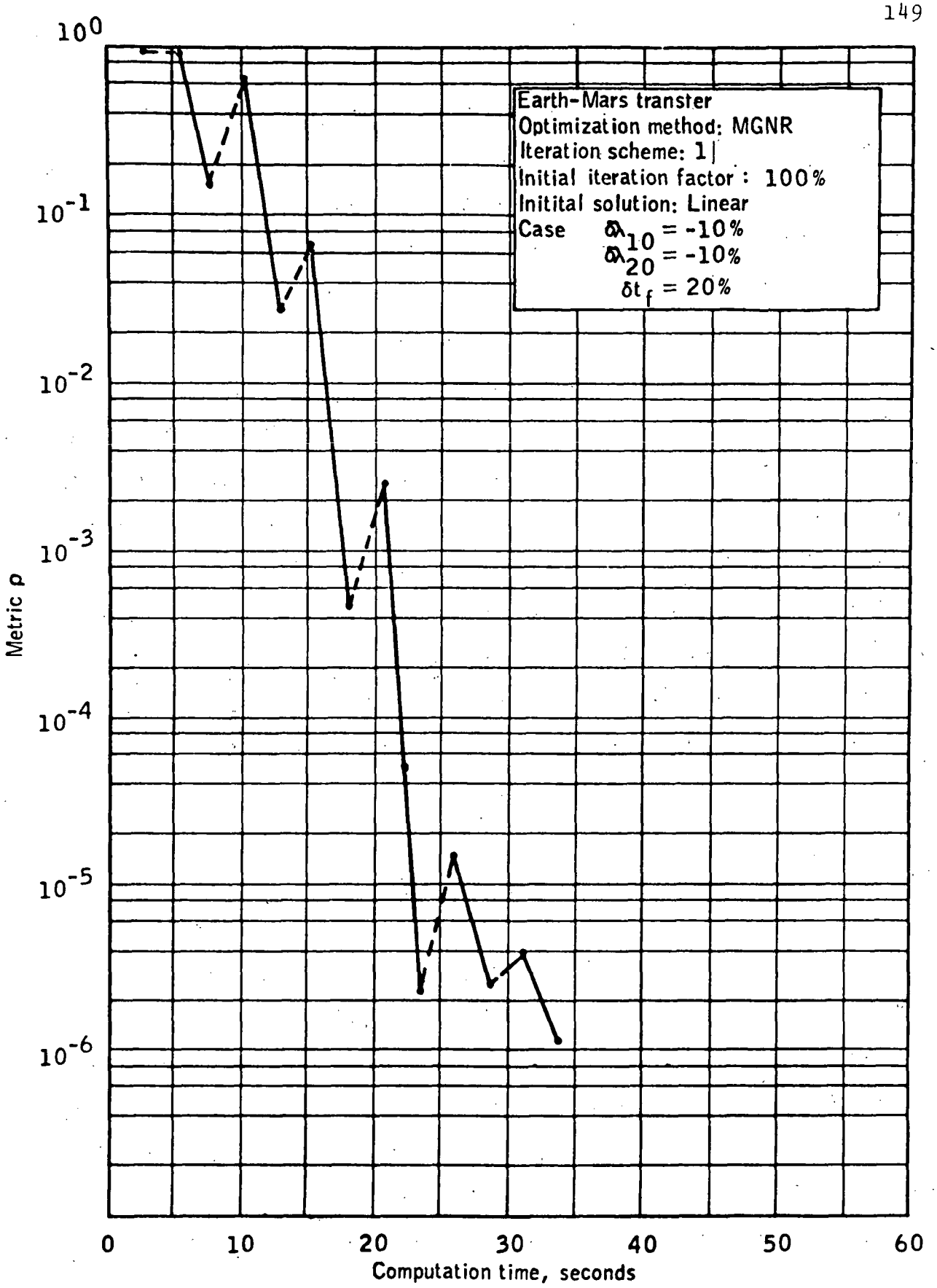


Figure 29. - Metric ρ as a function of convengence time for the MGNR using iteration scheme 1, initial iteration factor of 100% and a linear initial solution.

that the independent variable range is the same for all iterations, thus simplifying the programming slightly. A disadvantage is that one additional equation must be integrated and a rather complex term is added to each of the existing equations. The most significant advantage is that the terminal time determination becomes an integral part of the iteration process.

The convergence characteristics of the MMGNR is illustrated in Figure 30 for the same case shown in Figures 28 and 29 for the Normal Scheme and Iteration Scheme 1, respectively, using the MGNR. The metric reduction becomes a monotonic function of computation time, and when a linear initial solution is used the convergence time is 27 percent less than that required by the MGNR using the Normal Scheme. Figure 30 also shows the convergence characteristics for the case where the initial trial solution is determined from integrating the nonlinear differential equations.

6.4.2 Modified Quasilinearization Method

The required formulation of the Modified Quasilinearization Method as discussed in Appendix A.2.3 is simple and relatively easy to apply and this method is capable of handling terminal constraint functions. The terminal time determination is included as an integral part of the process and this method is very efficient compared to the MGNR. Also, no additional equations or terms are needed as with the MMGNR.

The programming effort requires the forward integration

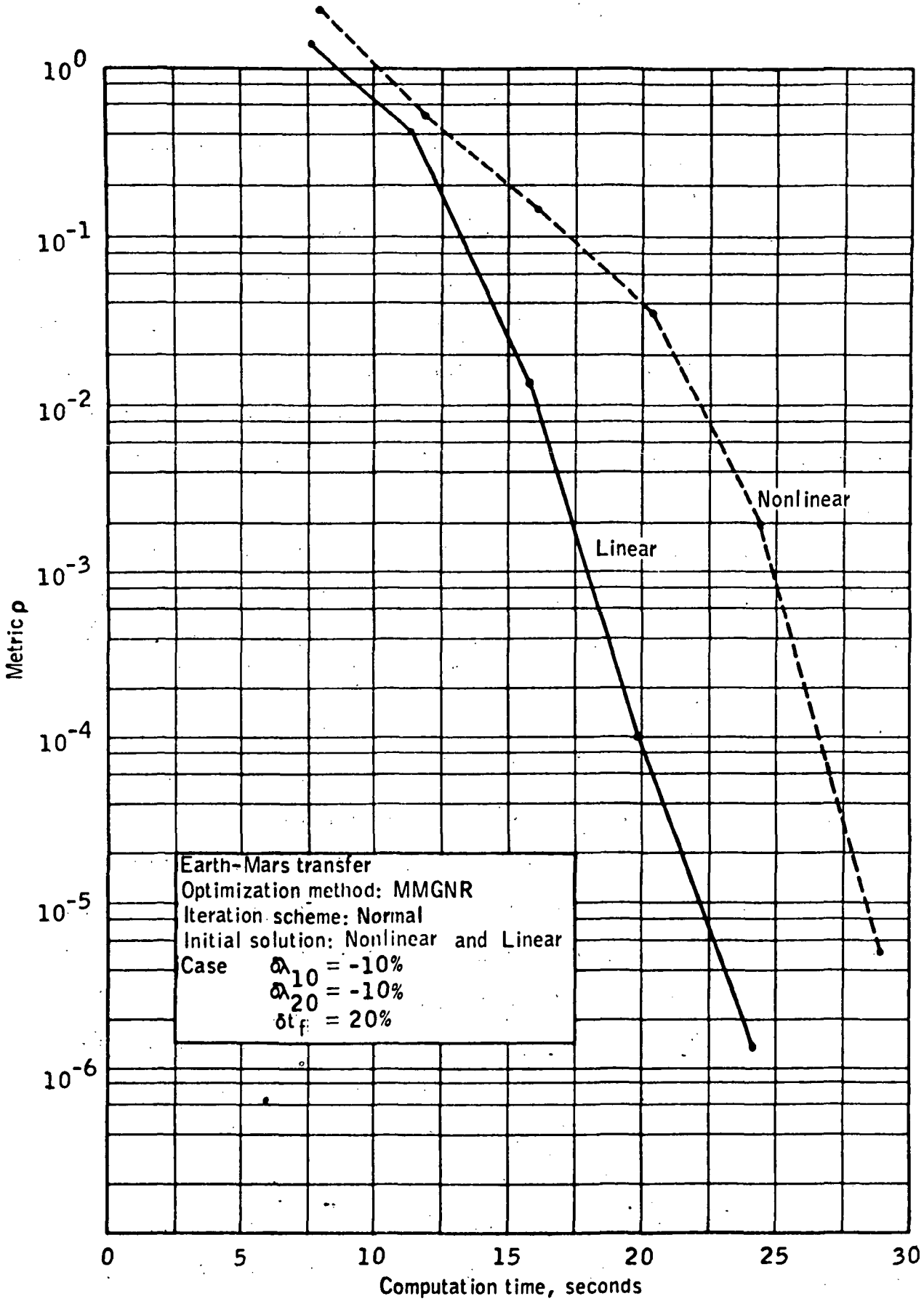


Figure 30. - Metric ρ as a function of computation time for the MMGNR using the normal iteration scheme.

of the homogeneous parts of eight linearized differential equations and Euler-Lagrange equations. Also, the nonhomogeneous parts are integrated forward and all coefficients are evaluated from the previous nominal. The corrections that must be applied for the next iteration are determined by solving a linear system. Only the terminal values of the forward integrations are stored as explained in Section 6.4.1. When a positive terminal correction is made, a linear extension of the variables from the previous nominal is made.

The computer program that uses the MQM requires two initially assumed Lagrange multipliers, an assumed terminal time, and an initial trial solution. In a manner similar to the MGNR, if a reasonable initial solution cannot be selected, the nonlinear equations may be integrated to provide an initial solution. The program is built such that only the subroutines containing the nonhomogeneous and homogeneous equations and the trial solution must be changed to solve different problems.

The Normal Scheme of the MQM is that of requesting a 100 percent correction in the terminal constraints. Iteration Scheme 2, used with the MQM, is similar to Iteration Scheme 1 for the MAF or MPF, where a percentage correction in the terminal constraints is requested. The logic required to determine whether the iteration factor is increased or decreased in the Quasilinearization Methods is more complex than that required for the MAF or MPF, because the metric ρ must be determined. This calculation requires several operations on all eight dependent

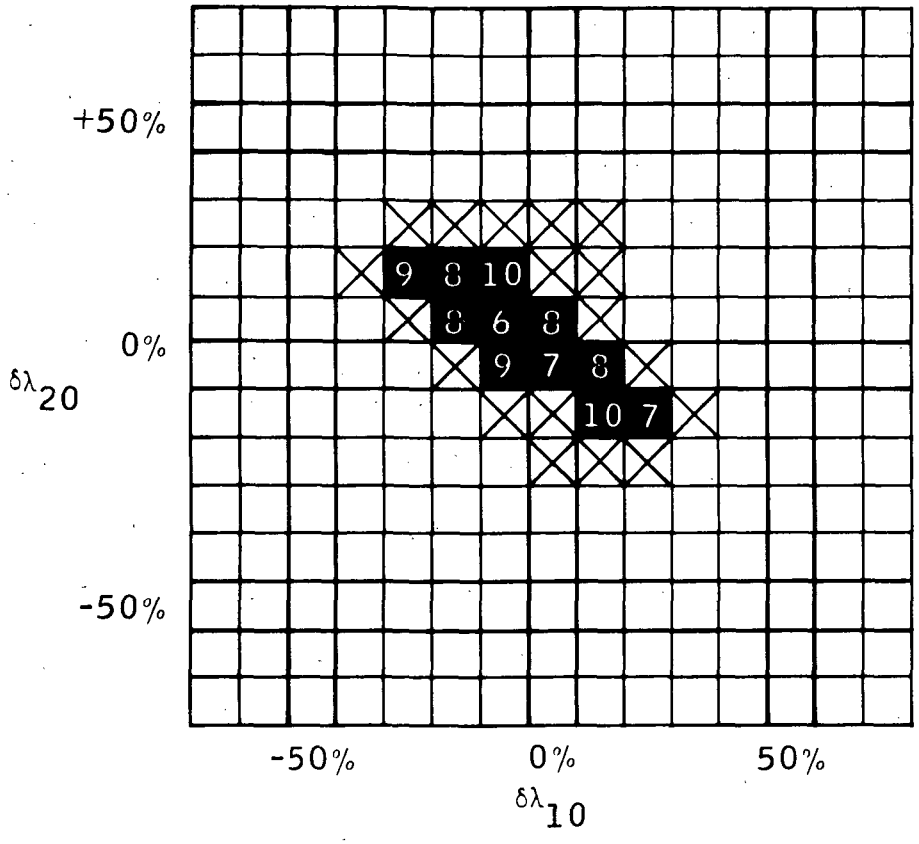
variables at each time step and hence requires a relatively large amount of time compared to the calculation of the norm in the MAF or MPF.

The convergence envelopes for the MQM using the Normal Scheme, a nonlinear initial trial solution and -20, 0, and 20 percent errors in terminal time, respectively, are shown in Figures 31, 32, and 33. The nonlinear initial trial solution is the one that results from integrating the nonlinear differential equations. Comparing these Figures with the Figures 1, 2, and 3 for the MAF reveals that while the general shape of the envelopes are the same, the MQM results in slightly smaller envelopes. For negative and zero terminal time errors, the method is extremely sensitive to Lagrange multiplier errors that have the same sign. For positive terminal time errors, the method is much more sensitive to positive λ_2 errors than to negative λ_2 errors.

An attempt to generate the same envelopes by using the MQM with a constant initial trial solution must be recorded as a failure, because no convergent solutions were obtained. The constant initial trial solution used is illustrated in Figure A.2.1 by short dashed lines.

Figures 34, 35, and 36 illustrate the convergence envelopes for MQM using Iteration Scheme 2 with an initial iteration factor of 50 percent, a nonlinear initial solution and -20, 0, and 20 percent errors in terminal time, respectively. These envelopes are significantly larger than the envelopes for the Normal Scheme shown in Figures 31, 32, and 33. It is interesting

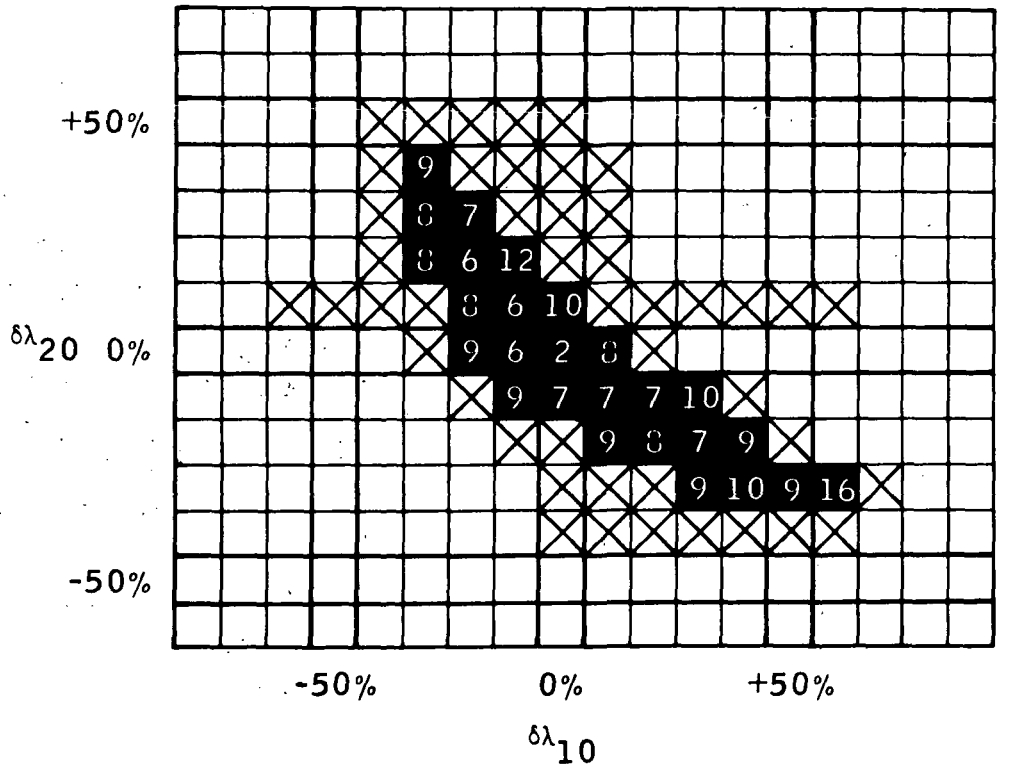
Earth-Mars transfer
Optimization method: MQM
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: -10%
Initial solution: Nonlinear



Note: The numbers indicate the iterations required for convergence

Figure 31.- Convergence envelope for the MQM using the normal iteration scheme, initial iteration factor of 100% and terminal time error of -20%.

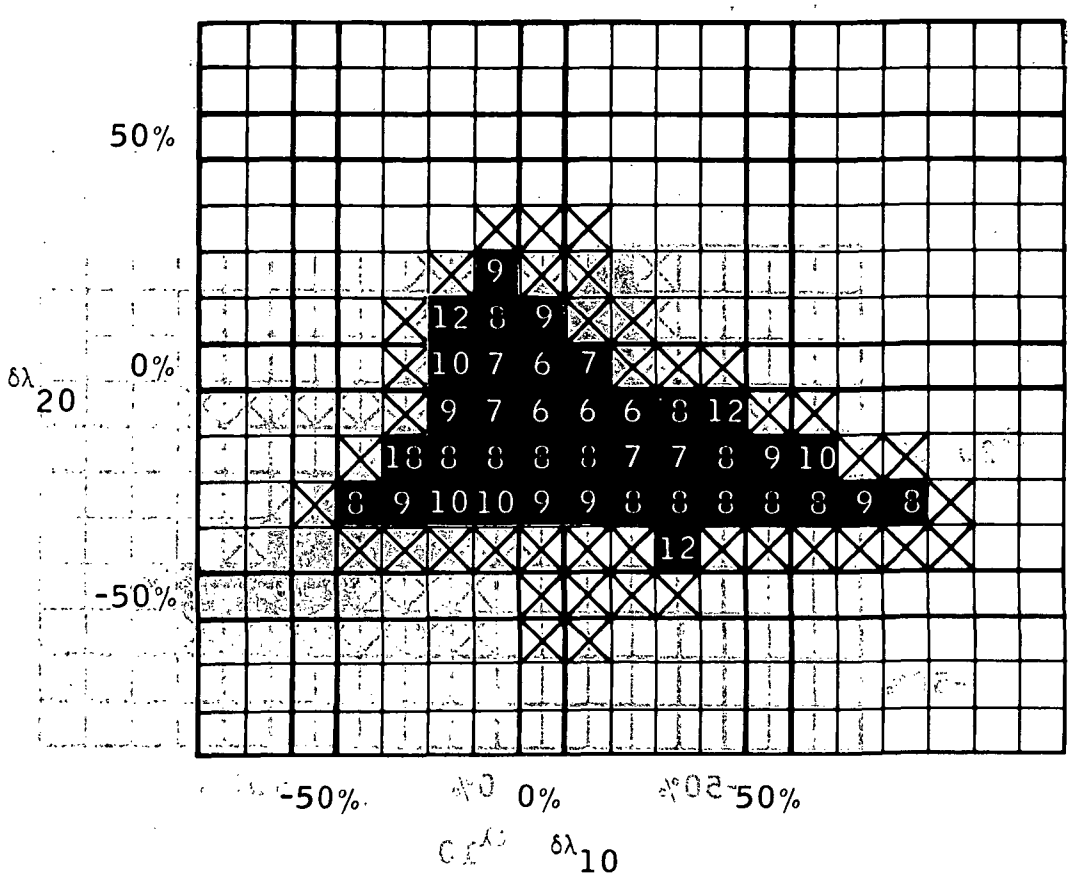
Earth-Mars transfer
Optimization method: MQM
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: 0%
Initial solution: Nonlinear



Note: The numbers indicate the iterations required for convergence

Figure 32.- Convergence envelope for the MQM using the normal iteration scheme, initial iteration factor of 100% and terminal time error of 0%.

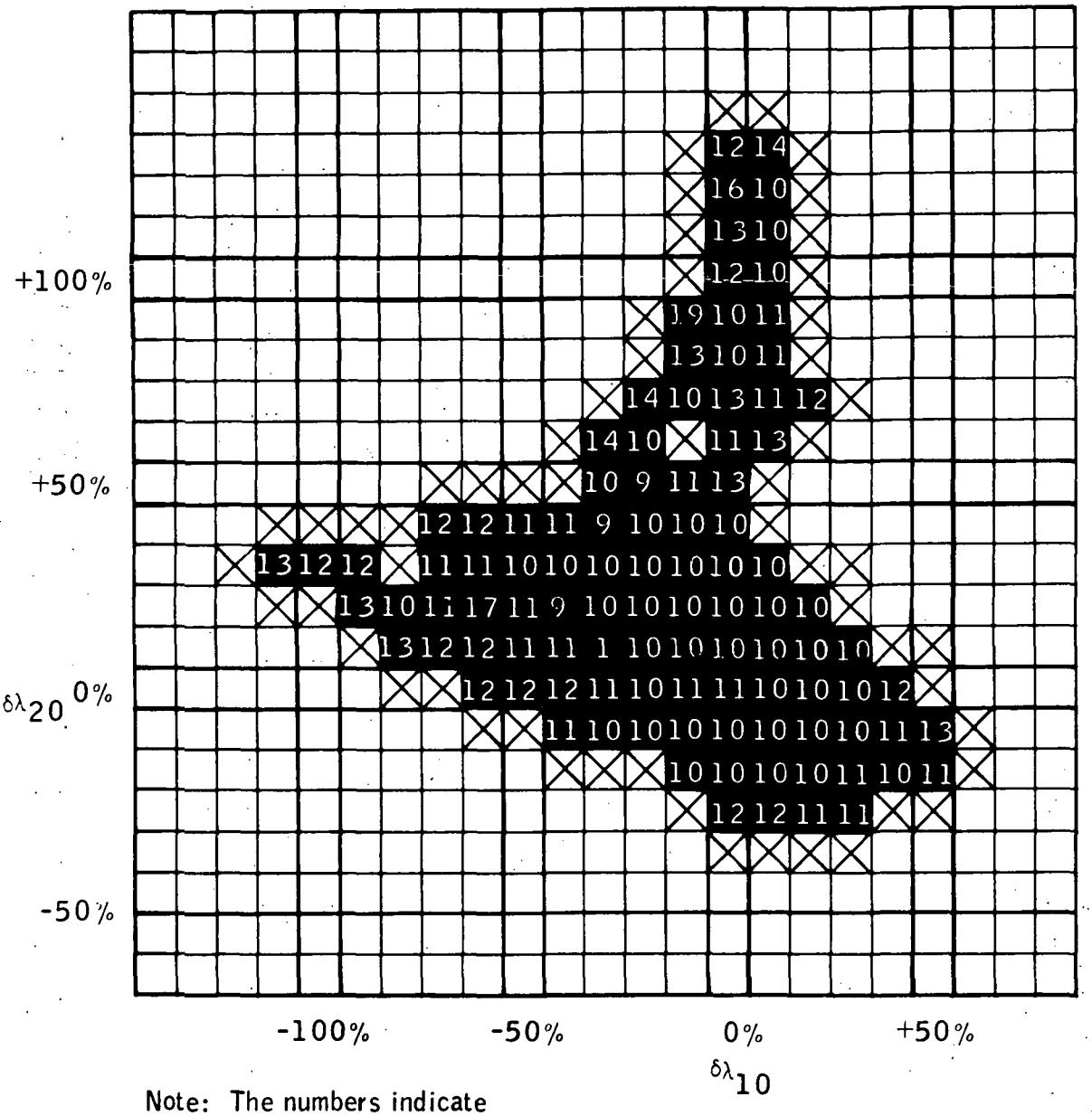
Earth-Mars transfer
Optimization method: MQM
Iteration scheme: Normal
Initial iteration factor: 100%
Terminal time error: 20%
Initial solution: Nonlinear



Note: The numbers indicate the iterations required for convergence

Figure 33.- Convergence envelope for the MQM using the normal iteration scheme, initial iteration factor of 100% and terminal time error of 20%.

Earth-Mars transfer
Optimization method: MQM
Iteration scheme: 2
Initial iteration factor: 50%
Terminal time error: -20%
Initial solution: Nonlinear



Note: The numbers indicate the iterations required for convergence

Figure 34.- Convergence envelope for the MQM using iteration scheme 2, initial iteration factor of 50% and terminal time error of -20%.

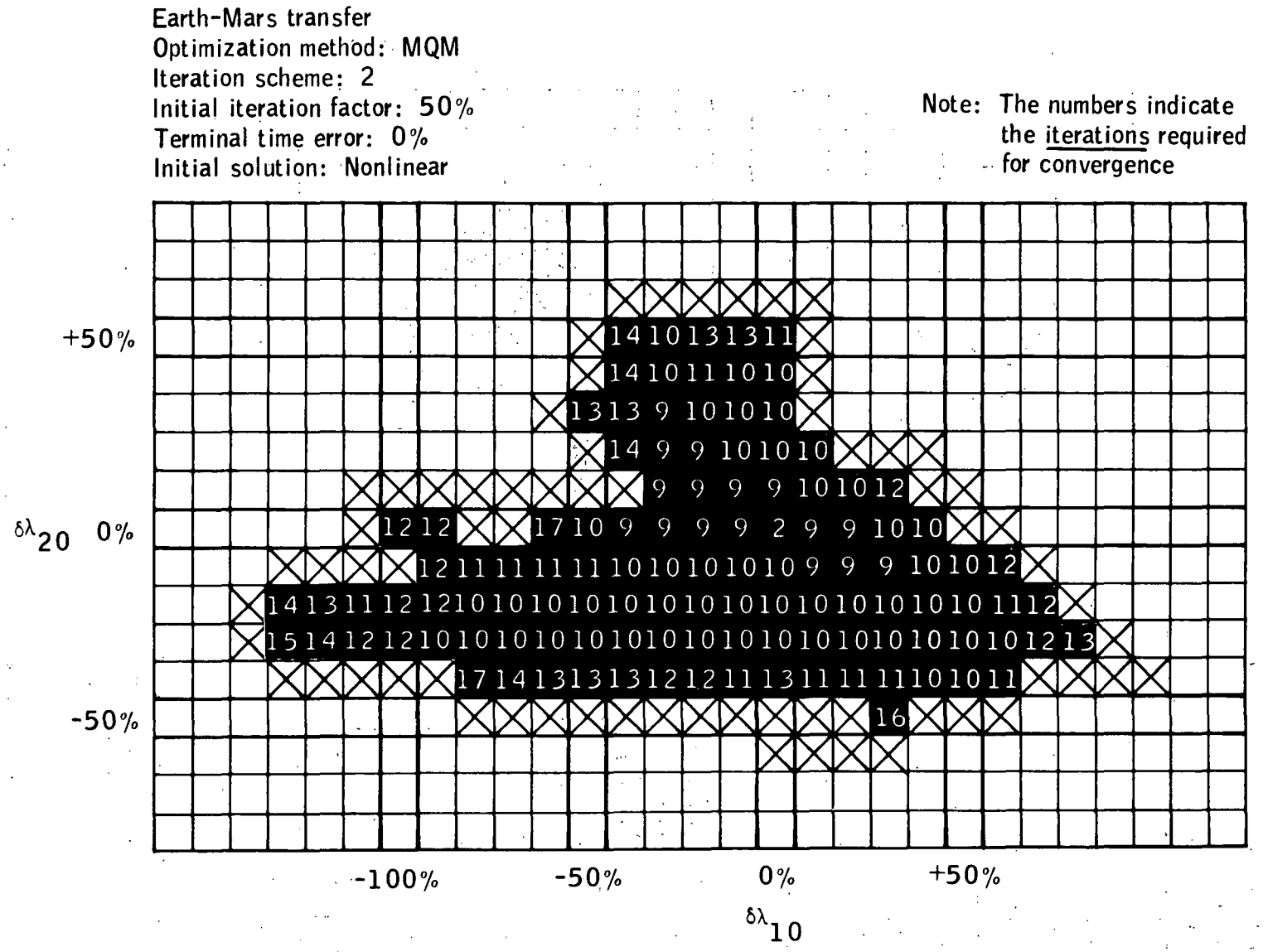


Figure 35.- Convergence envelope for the MQM using iteration scheme 2, initial iteration factor of 50% and terminal time error of 0%.

Earth-Mars transfer
Optimization method: MQM
Iteration scheme: 2
Initial iteration factor: 50%
Terminal time error: 20%
Initial solution: Nonlinear

Note: The numbers indicate the iterations required for convergence

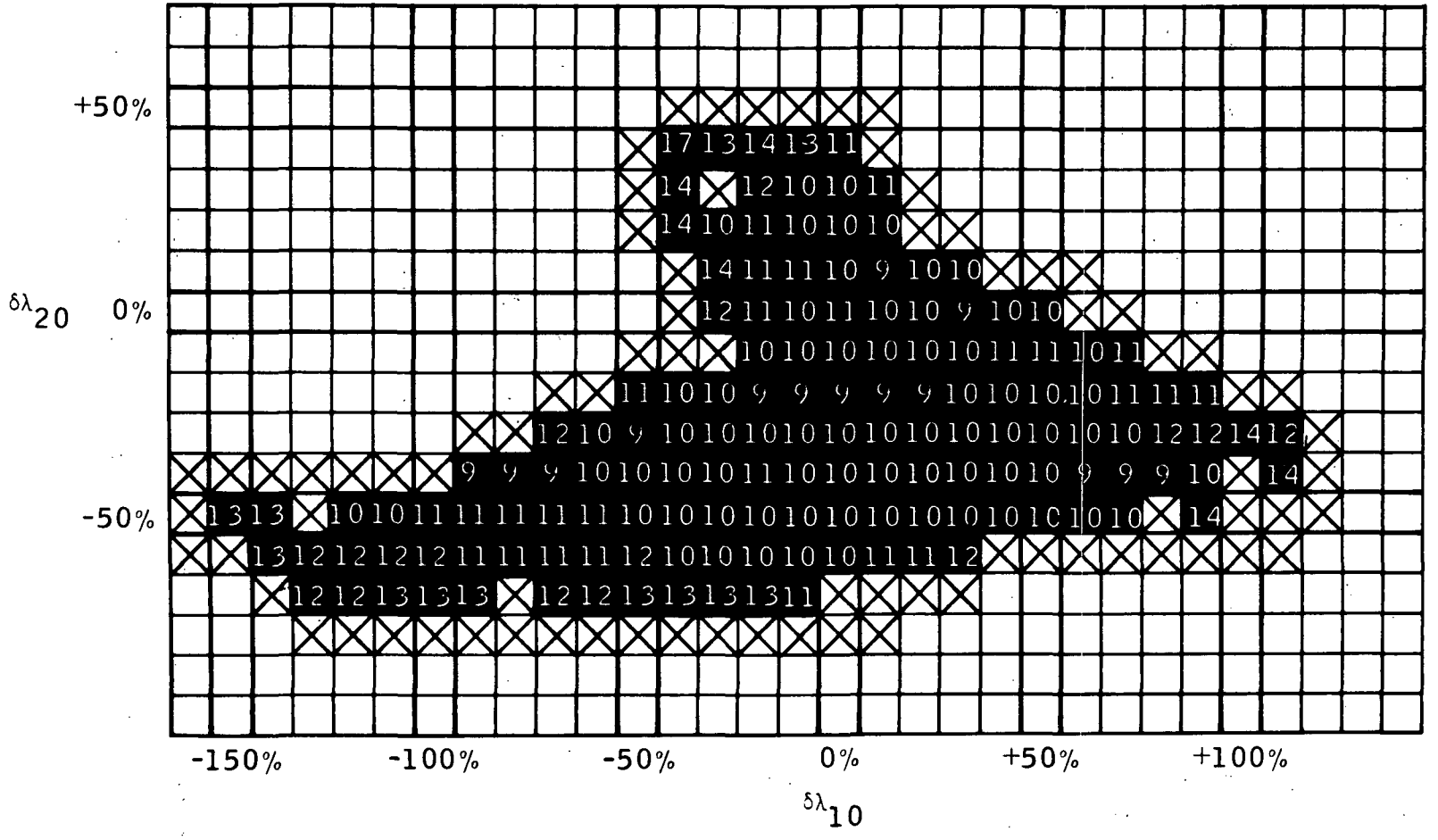


Figure 36.- Convergence envelope for the MQM using iteration scheme 2, initial iteration factor of 50% and terminal time error of 20%.

to note that while the envelopes for the Normal Scheme are slightly smaller than the corresponding envelopes for the MAF, the envelopes shown in Figures 34, 35, and 36 are slightly larger than the corresponding envelopes for the MAF shown in Figures 7, 8, and 9. This suggests that Iteration Scheme 2 for the Quasi-linearization Methods is more effective than Iteration Scheme 1 for the Perturbation Methods. The Figures 34, 35, and 36 follow the pattern previously mentioned for the other methods in that the method is increasingly sensitive to positive λ_2 errors as the terminal time error increases.

It is of definite interest to note the required convergence times for the cases illustrated for the MQM. As an example, Figure 37 shows the convergence times for the envelope of Figure 35. This envelope may be compared directly with the corresponding envelopes generated by the MAF in Figures 15, 16, 17, and 18 and the MPF in Figure 27. An obvious fact is that the MQM requires slightly more computation time than the MAF and MPF, but shows considerable improvement over previous quasilinearization techniques such as the MGNR and MMGNR. In all fairness, however, it must be pointed out that more time was spent in trying to make the programming efficient for the MQM than for the MGNR and MGNRM.

An insight to the convergence characteristics of the MQM may be seen in Figure 38 for the special case where the Lagrange multiplier and terminal time variations are -10, -10, and 20 percent, respectively. This figure may be compared directly with

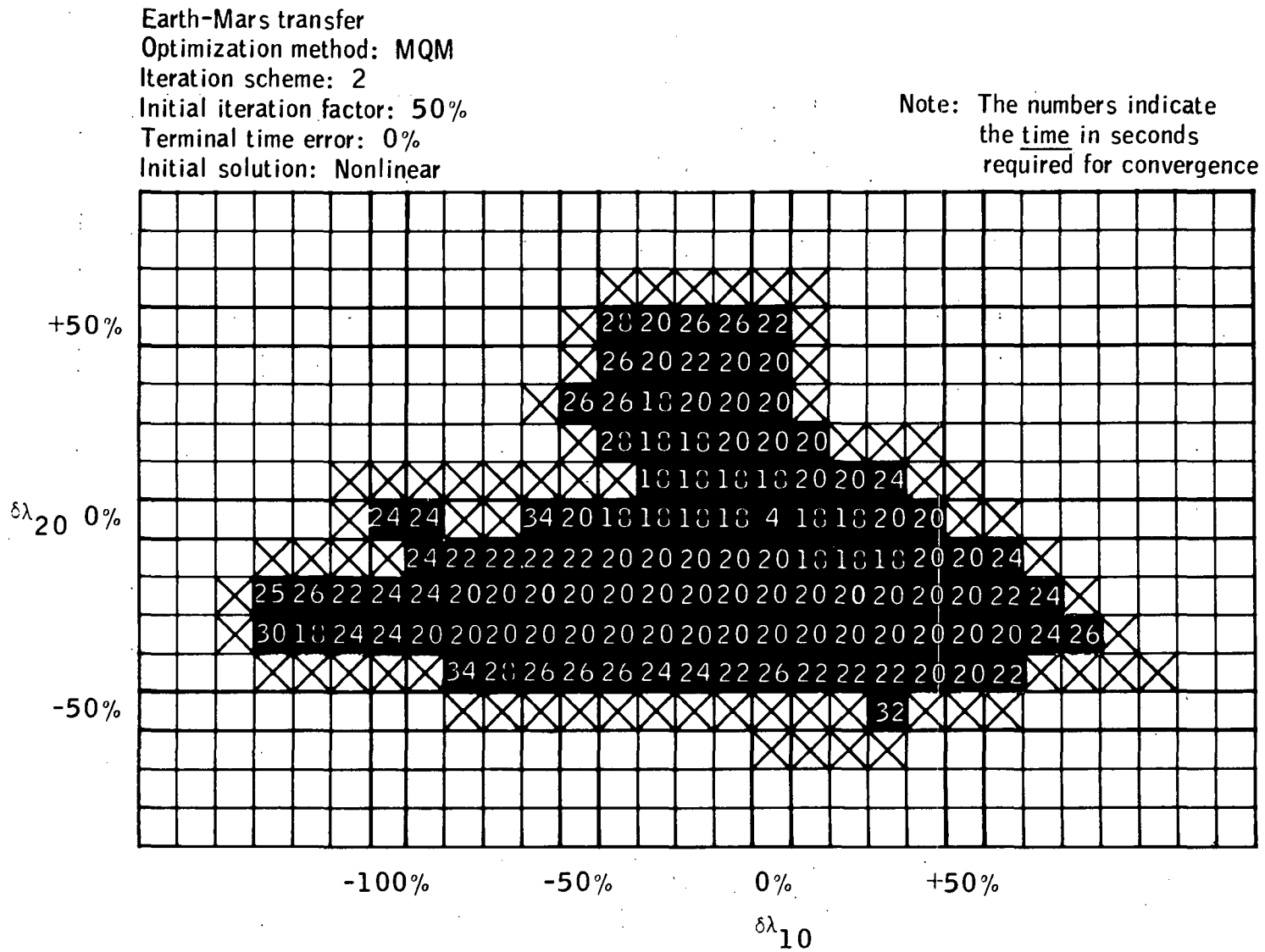


Figure 37.- Convergence envelope for the MQM using iteration scheme 2, initial iteration factor of 50% and terminal time error of 0%.

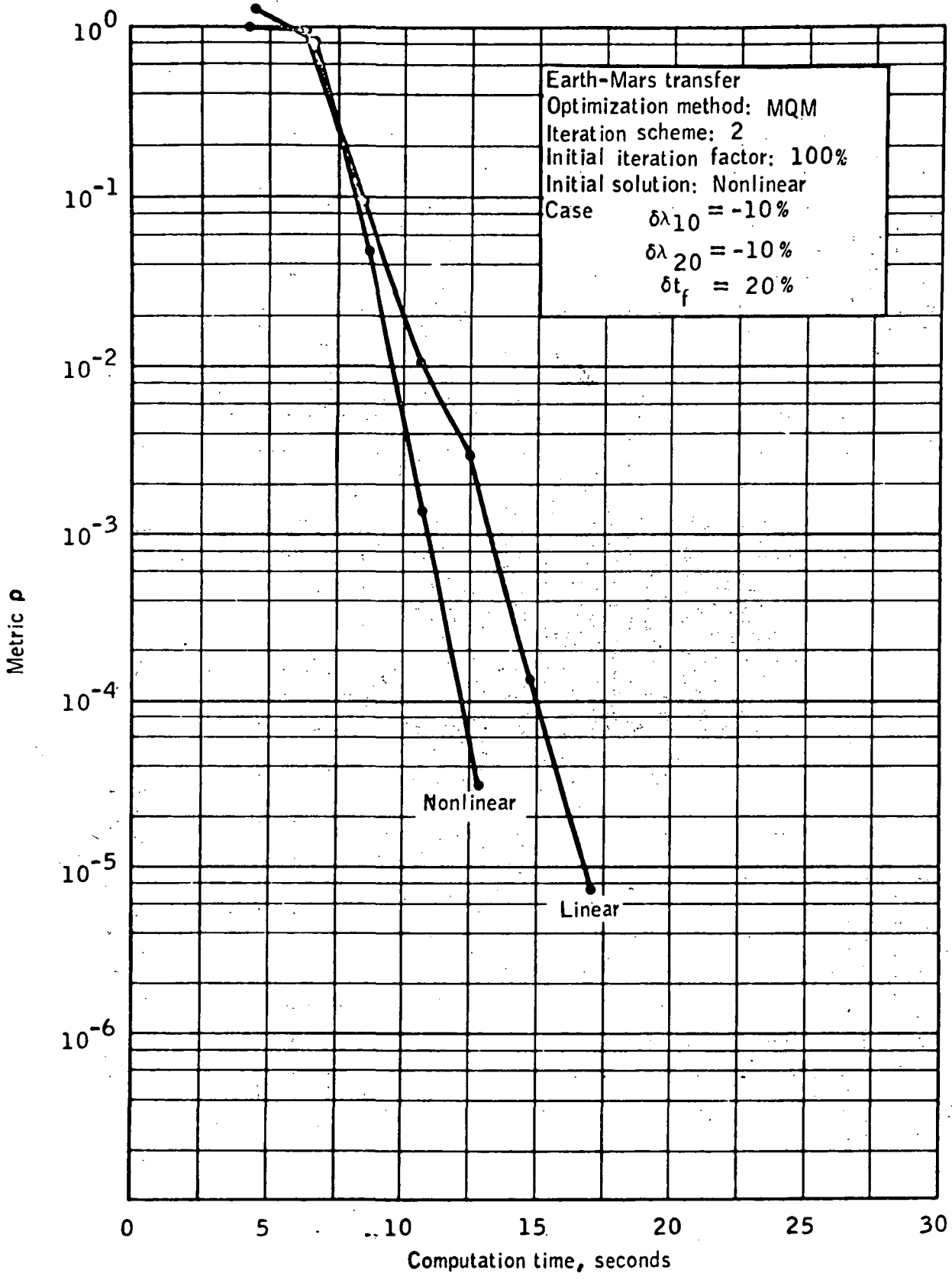


Figure 38. - Metric ρ as a function of computation time for the MQM using iteration scheme 2, and initial iteration factor of 100%.

Figures 28, 29, and 30 for the MGNR using the Normal Scheme, MGNR using Iteration Scheme 1 and MMGNR, respectively. Figure 38 may also be compared, in a sense, with the 100 percent curve in Figure 19 for the MAF. Caution must be exercised, however, because the ordinates represent different quantities. It is expected that a reduction of the metric ρ is more stringent a requirement than reduction of the terminal constraint norm. The more stringent requirement results from the fact that the metric ρ is composed of so much more information than the terminal constraint norm.

Figure 39 illustrates the effect of the initial value of iteration factor on convergence time for two specific cases of initial parameter error. This figure may be compared to Figure 13 which represents the same information for the MAF for the same cases. The same characteristics are noted in that for some cases the best initial iteration factor is somewhat less than 100 percent and that this best value is not the same for all cases. One additional characteristic, noted in Figure 39, is that very large penalties in the convergence times are paid when low initial iteration factors are used. This deficiency is attributed to the metric criteria used to determine how the iteration factor must be changed. When only a small percentage correction is requested, the metric does not decrease rapidly at first. This is because the metric is interpreted as the maximum distance between successive trajectories. In fact, in application the metric sometimes increases slightly and this causes the iteration factor

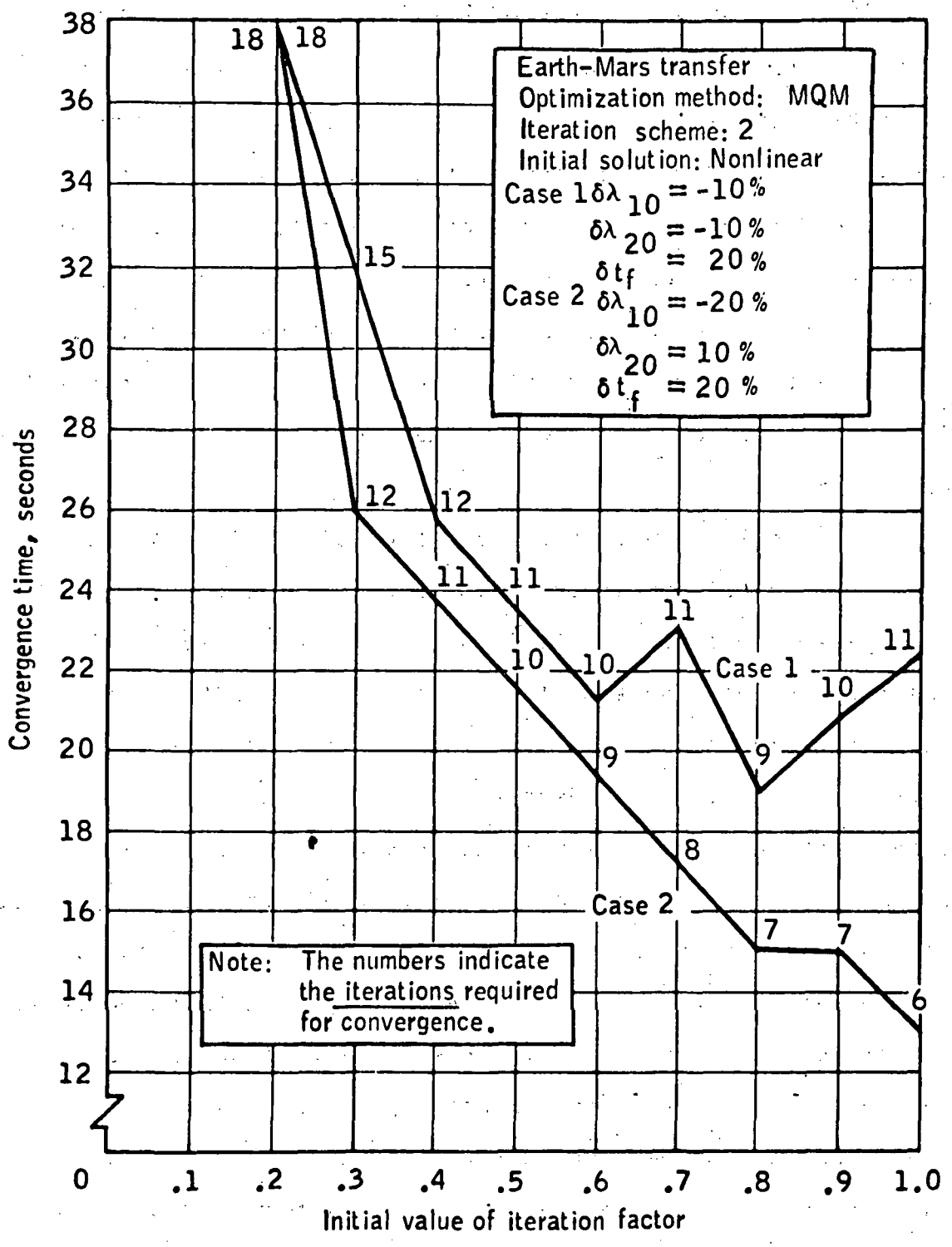


Figure 39. - Convergence time as a function of the initial value of iteration factor for the MQM using iteration scheme 2, and a nonlinear initial solution.

to decrease. This process could conceivably have a decelerating effect on the convergence. This phenomena may be seen in Figure 40 for the case where the initial iteration factor is 20 percent.

Figure 40 also illustrates the convergence characteristics for several different initial iteration factors and may be compared to Figure 19 which represents the same information for the MAF for the same case. It should be noted that near the terminal phase of each trial the metric reduction is nearly quadratic.

In summary, the Quasilinearization Methods show a wide range of convergence characteristics, but the proposed method, the MQM, successfully reduces the convergence times and increases the convergence envelopes to become competitive with the MAF and MPF.

Generally speaking, the MQM displays the same characteristics that are seen for the MAF and MPF. For the case when an initial iteration factor of 50 percent is used, the envelope of convergence for the MQM is slightly larger than the corresponding envelope for the MAF and MPF. But the convergence times are always slightly larger than for the MAF.

6.5 Gradient Methods

The comparison and discussion of the Gradient Methods will consist of two separate analyses. The Method of Steepest Descent, including Iteration Schemes 1 and 2, is discussed first, and the Modified Method of Steepest Descent is discussed last.

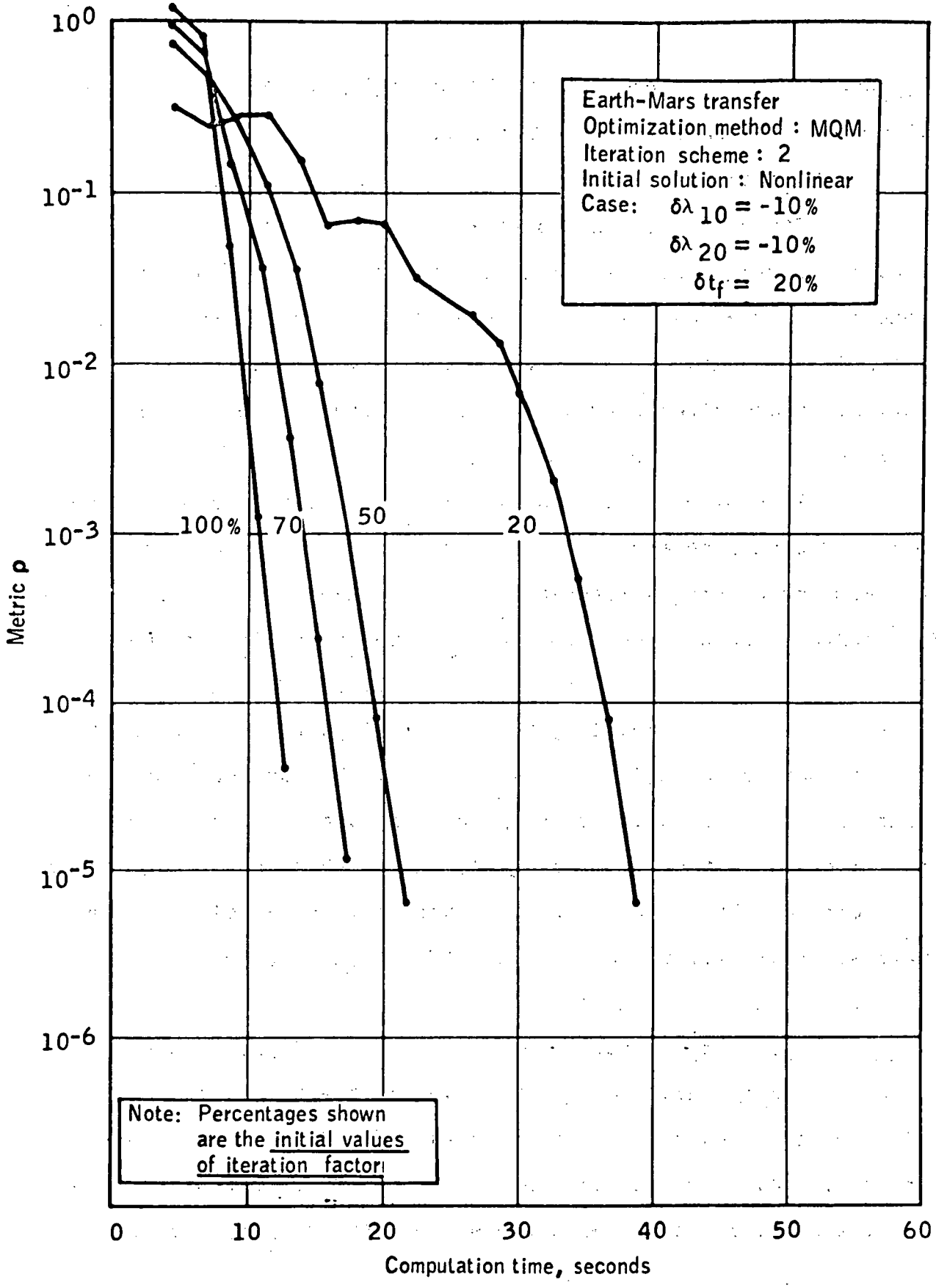


Figure 40. - Metric ρ as a function of computation time for MQM using iteration scheme 2 and a nonlinear initial solution.

The discussion content will include the applicable items listed in Section 6.2.

6.5.1 Method of Steepest Descent

The required formulation of the Method of Steepest descent as discussed in Section 5.1 is simple and straightforward, but slightly cumbersome when compared to the MAF or MPF. A specific application of the MSD is presented in Appendix A.2.4.

The programming effort requires forward integration of four differential equations of motion, storing the dependent variables in computer memory or on tape at each time step. This requires less storage than storing the A and B matrices. The four adjoint differential equations, Eq. (5.6), are integrated backwards five times using the variables stored during the forward integration to form the coefficients. One additional complexity is that Eqs. (5.30) through (5.32) must also be integrated backwards, and may be carried along simultaneously with the adjoint equations. To reduce the programming complexity, a constant integration step is used for all integrations. The computer storage problem can be eliminated by integrating the differential equations of motion backward along with the adjoint equations, Eq. (5.6), and Eqs. (5.30) through (5.32). This is not done in the present method because the equations of motion must be integrated forward anyway to determine the terminal values of state.

In addition to the programming effort explained above,

the most serious disadvantage of the MSD is that a moderate amount of human intervention and experience is required to implement the program. For example, the weighting matrix W is not defined, and by just using the unity matrix the less sensitive regions of the control program are very slow in acquiring the optimal shape. The weighting matrix may be used to speed this optimal shaping process, but the insensitive regions of the control program are not always known.

An examination of Eq. (5.33) reveals that the first group of terms are related to the minimizing effort while the last group of terms are related to the terminal constraint satisfaction. There is, however, some cross coupling of the terminal constraint satisfaction in the first term. The procedure used to affect convergence requires a selection of an allowable average control deviation, based on Eq. (5.19), that does not invalidate the linearity constraints on the problem. This allowable control deviation must be reduced in some specified manner as the process progresses. If the numerator of the radical in Eq. (5.33) is negative when 100 percent correction in the terminal dissatisfaction is requested, the percent correction that causes the radical term to vanish is determined. When this occurs, emphasis is placed on reducing the terminal dissatisfaction. If the numerator is positive when 100 percent correction is requested, the radical is used and both the performance index is reduced, and the terminal constraints are driven toward satisfaction. The computer logic involved in the above operations requires a

significant amount of the iteration time.

The computer program that uses the MSD requires the initial value of the state variables, a stopping condition and an assumed control program. These estimates require some familiarity with the physical problem. The stopping condition that is chosen must be one that will be satisfied. The control program selection is not as critical as it is for the MQM. The computer program is not so easily generalized as it is for the MAF, MPF, or MQM, i.e., extensive programming is required to accommodate a different problem.

Iteration Scheme 1 simply uses the unity matrix for W and Iteration Scheme 2 uses the H_{uu}^{*} matrix. This second scheme requires some additional computer storage and programming. When Iteration Scheme 2 is to be used, H_{uu}^{*} must be formed with the variables that result from integrating the adjoint differential equations backwards, using ν as given in Eq. (5.27) for the starting conditions. A major problem when using Iteration Scheme 2 is that when a percentage correction in the terminal constraints is requested, thereby forcing the radical term in Eq. (5.33) to vanish, ν becomes infinite. Clearly this cannot be used as a starting condition for the adjoint equations.

With the examples discussed, this radical term vanishes for the first few iterations, and when this happens the unity weighting matrix is used. As soon as the radical becomes finite, the H_{uu}^{*} matrix is calculated for use on the following trajectory.

The thrust angle as a function of mission time for the Earth-Mars transfer is shown in Figures 41, 42, 43, and 44, and the convergence process from the assumed history to the Eulerian history is illustrated. Figures 41 and 42 show the convergence characteristics for two widely different initially assumed control programs, designated Case 1 and Case 2, using Iteration Scheme 1. It is interesting to note that the number of iterations required is relatively independent of the initial control program. After 30 iterations both cases yield control programs that almost obscure large portions of the Eulerian program, and hence are not shown. When to terminate the iteration process is not clear since the Eulerian optimal is really never reached. The method used here was to continue until no further improvement was being made, i.e., until the solution began to oscillate about some mean path. A more sophisticated method would be to terminate when a time integral of H_u or $H_u{}^T H_u$ became arbitrarily small.

An apparent discontinuity begins to develop at approximately 100 days, as seen in Figure 41, and becomes more severe as the iterations progress. After 30 iterations the apparent discontinuity becomes very sharp and the Eulerian control is accurately approximated. The same characteristic is noted in Figure 42.

The effectiveness of Iteration Scheme 2 in shaping the optimal control program is illustrated in Figures 43 and 44, and it is seen that the number of iterations required is significantly reduced. In comparing Figures 41 and 43, for instance,

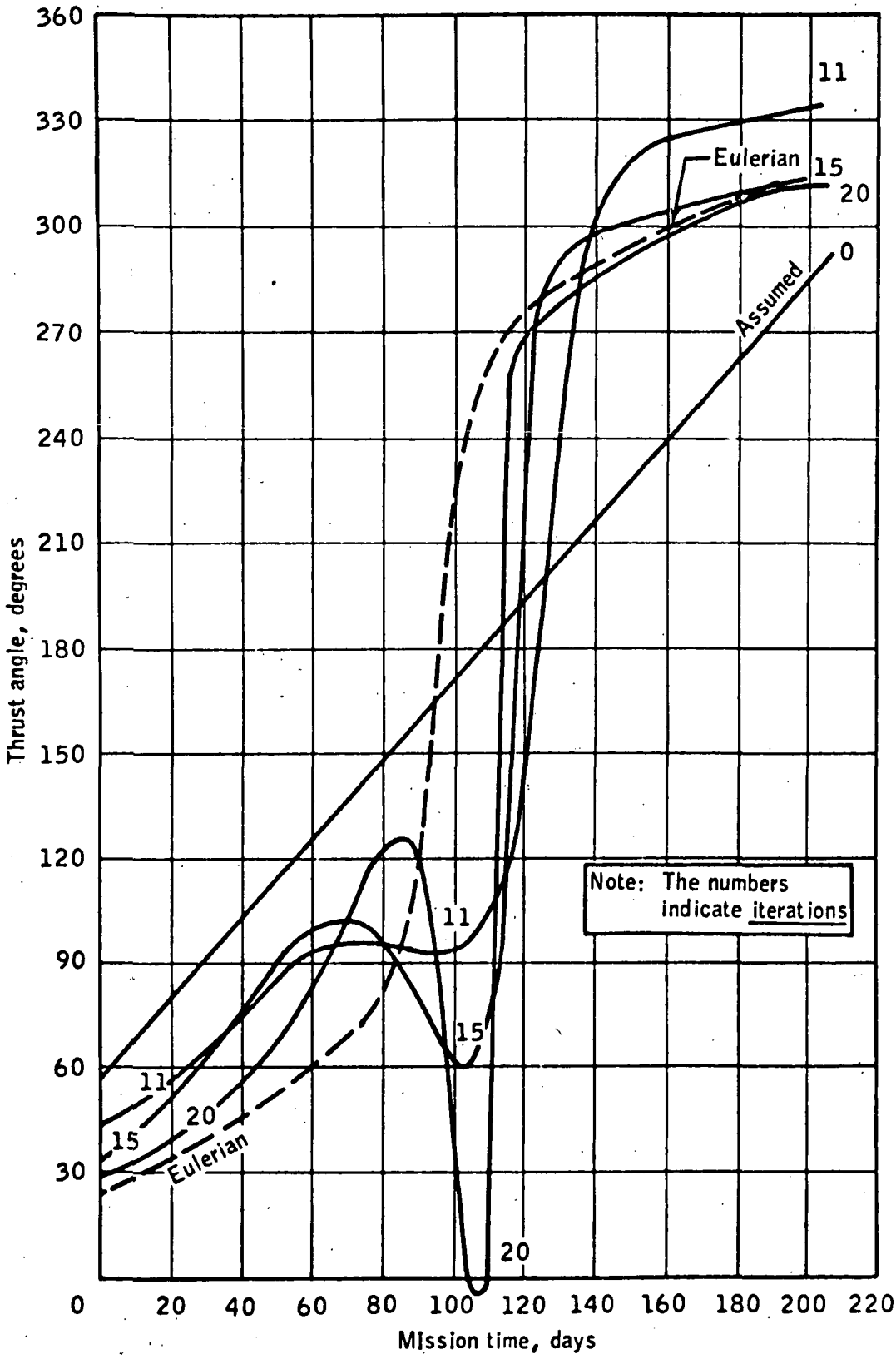


Figure 41. - Thrust angle as a function of mission time for Earth-Mars transfer using the MSD and weighting matrix W = I.

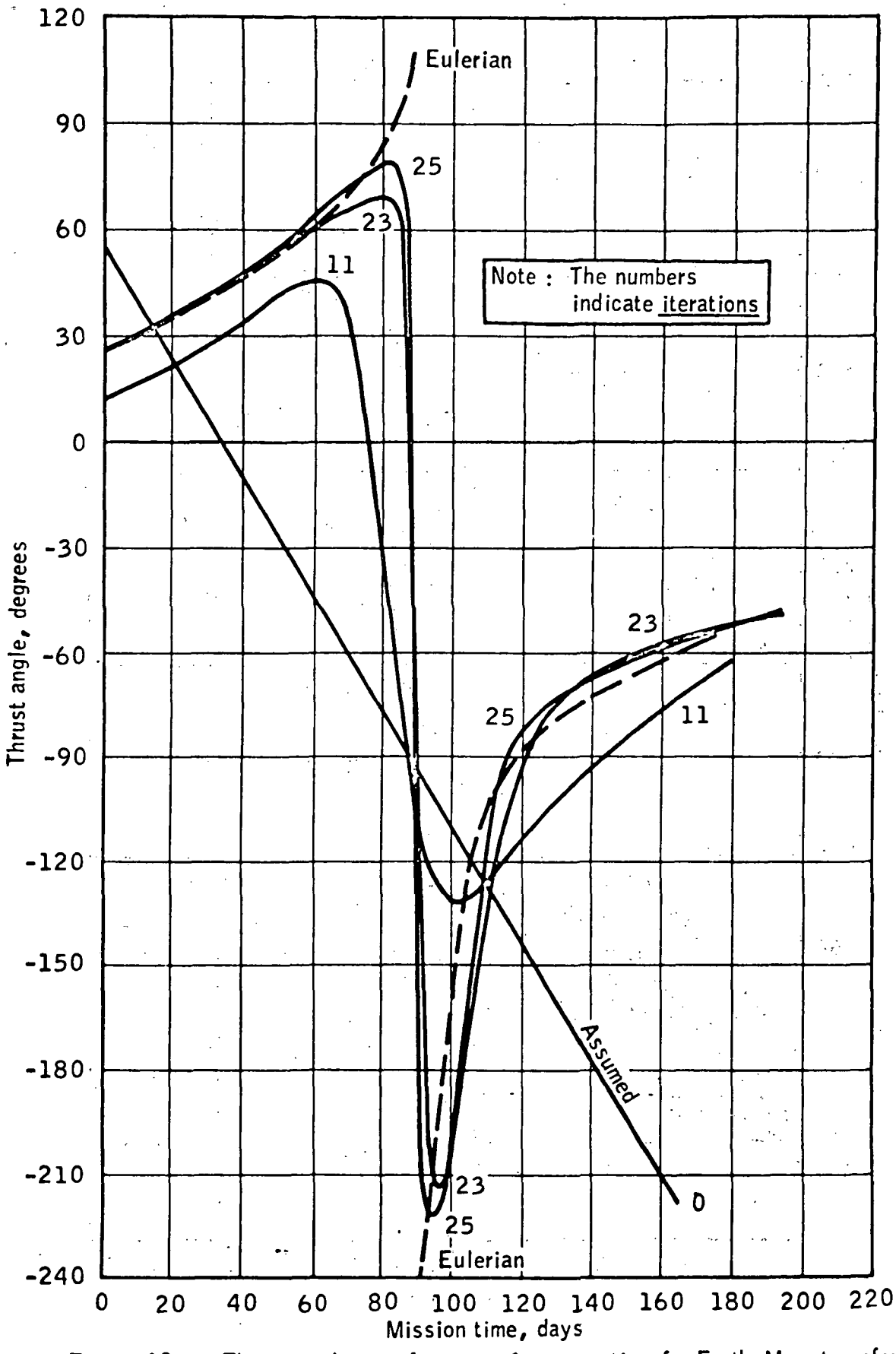


Figure 42. - Thrust angle as a function of mission time for Earth-Mars transfer using the MSD and weighting matrix W = I.

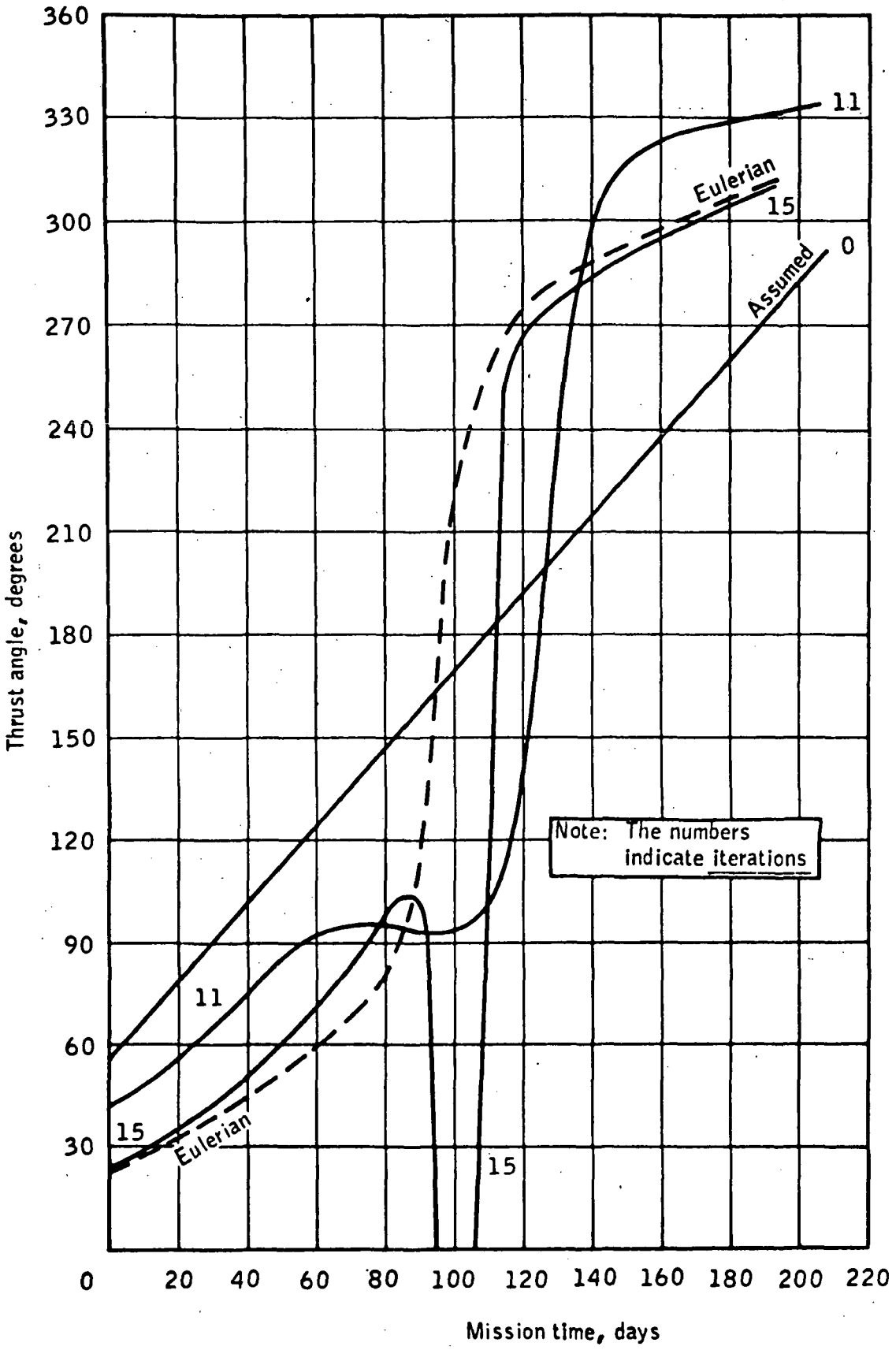


Figure 43. - Thrust angle as a function of mission time for Earth-Mars transfer using the MSD and weighting matrix $W = H^{*}_{uu}$.

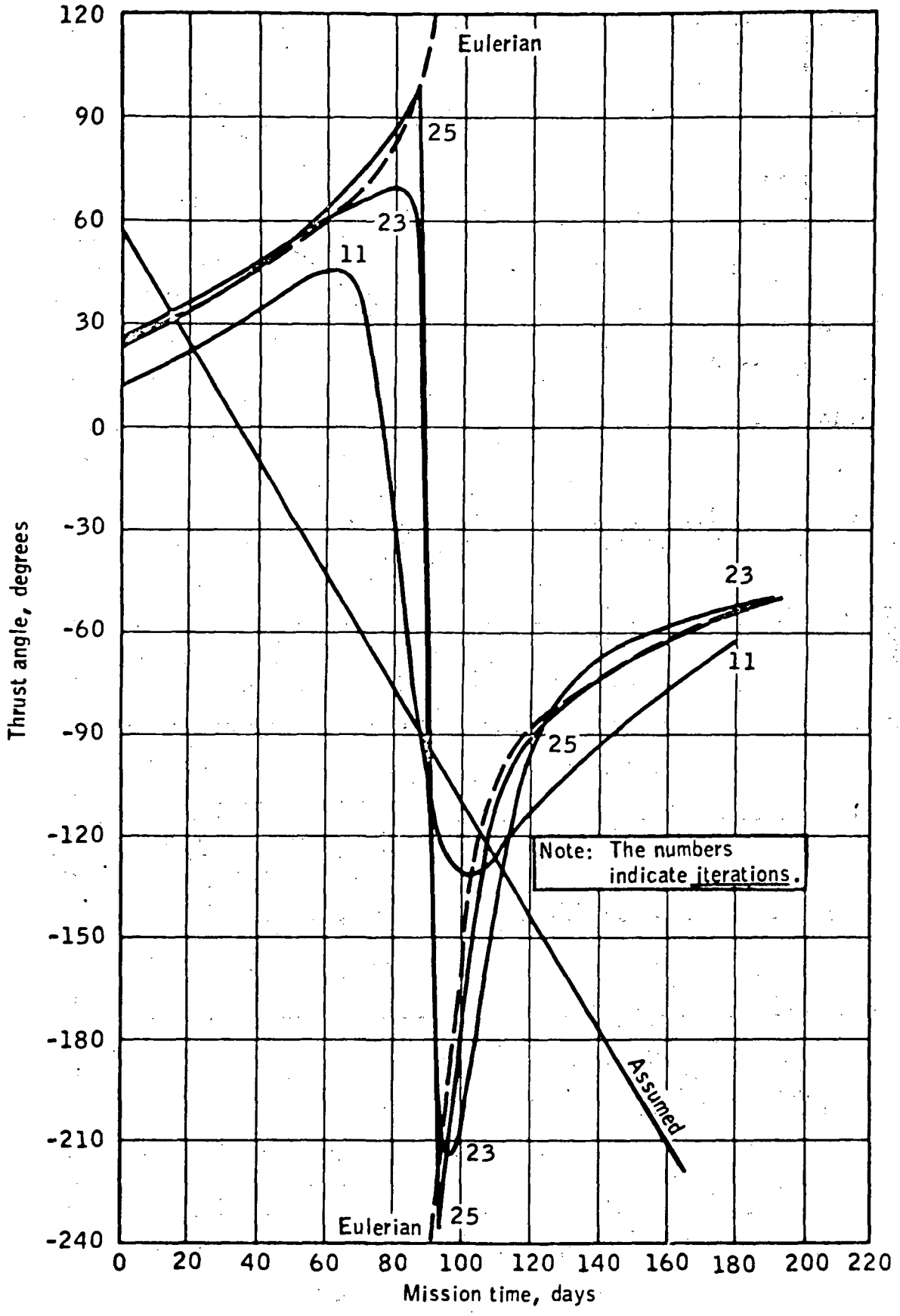


Figure 44. - Thrust angle as a function of mission time for Earth-Mars transfer using the MSD and weighting matrix $W = H^{*}_{uu}$.

it is seen that the apparent discontinuity development is much faster in the latter figure. These two cases are identical for the first 11 iterations because the radical in Eq. (5.33) vanishes and $W = 1$, but starting with the 12^{th} iteration, the H_{uu}^{*} matrix is formed and used. It is during these final iterations that the full value of Iteration Scheme 2 becomes evident. After only four additional iterations the apparent discontinuity, as shown in Figure 43, is well beyond the development shown in Figure 41. Moreover, the Eulerian is much better approximated, for a given number of iterations, when Iteration Scheme 2 is used.

The same characteristics are seen in Figures 42 and 44. For this case, however, the H_{uu}^{*} matrix is not calculated until the 23^{rd} iteration. After only two additional iterations, Iteration Scheme 2 in Figure 44 shows marked improvement in the development of the apparent discontinuity.

An average iteration for Iteration Scheme 1 requires approximately 2.75 seconds of computer time, while approximately 3.0 seconds is required with Iteration Scheme 2 when the H_{uu}^{*} matrix must be formed. However, an extensive step size study was not made for the MSD. The step size used was the same as that used for the integrations in the indirect methods.

It should also be pointed out that the terminal condition resulting from Eq. (2.14) may be used to determine the terminal value of the Lagrange multipliers. These values are used to start the backward integration of the adjoint equations,

for the H_{uu}^{*} determination, and also may be used to estimate the Lagrange multipliers required for starting the indirect optimization methods. For the case illustrated in Figure 44 the first time H_{uu}^{*} is determined, the values of λ_{10} and λ_{20} are calculated to be 2.15 and 0.65 percent larger than the values that correspond to the optimal trajectory, respectively. This error is well within the envelope of convergence of all the indirect methods studied.

6.5.2 Modified Method of Steepest Descent

The required formulation of the Modified Method of Steepest Descent as discussed in Section 5.2 is simple and straightforward, and is not as cumbersome as the MSD. A specific application of the MMSD is presented in Appendix A.2.5.

The programming effort requires forward integration of four differential equations of motion, storing the dependent variables in computer memory or on tape at each time step. This requires less storage than storing the A and B matrices. The four adjoint differential equations are integrated backward only once, using the variables stored during the forward integration. The Eq. (5.48) must also be integrated so that after a desired penalty function decrease is specified, a step size K may be determined. The MMSD requires a significantly reduced number of operations, as opposed to the MSD, because the adjoint equation is integrated backwards with three less starting vectors and the integration of Eq. (5.48) is much less time consuming than the

integration of Eqs. (5.30) through (5.32) in the MSD. The storage problem associated with the first forward integration may be avoided in a manner similar to that suggested in Section 6.5.1. The present method does store the forward integration and use a constant integration step size for all integrations.

In addition to the programming effort explained above, the most serious disadvantage of the MMSD is that a considerable amount of human intervention and experience is required to implement the program, even more than that required for the MSD. For example, the step size K is not defined, and must be approximated by using Eq. (5.48). A still more serious deficiency is that a constraint on the control deviation is not included as an integral part of the method itself, and hence appropriate computer logic must be used to insure that the linear constraints of the problem are not violated. One further complexity is that the convergence characteristics are highly dependent on the factors that weight the terminal constraints in the penalty function, and the magnitude of these factors are not specified. To compound the matter, the rates at which these factors are changed to tighten the terminal constraints are not known. It is seen that the price that must be paid for the simplicity of the method is that of increased arbitrariness, and a considerable amount of skill and experience is required to obtain meaningful results. This method has been programmed and is in the stage of evaluation, but no results are presented here.

6.6 Summary of the Comparison

The comparison of optimization methods thus far has consisted of individual analysis of each method with an occasional comment concerning the relative merits of one method with respect to the others. It would be helpful to summarize the conclusions of the comparison with particular emphasis on the basis of comparison as outlined in Section 6.2. A summary of the comparison is:

(1) The programming complexity and required formulation time is greater for the MQM and MSD than for the MAF, MPF and MMSD, because more computer logic is required.

(2) The MAF and MSD requires more computer storage than the other methods.

(3) The MSD and MMSD require more human intervention and intuition than the other methods, and hence are difficult for inexperienced personnel to use. However, the indirect methods become difficult to implement when the problem dimension is large.

(4) The computer program for the MSD requires considerable modification for solving a different problem, while the other programs require less modification.

(5) The convergence envelope sizes for all the indirect methods are essentially the same when the initial iteration factor is near 100 percent. The MQM envelope is

slightly larger than the envelopes of the other methods when the initial iteration factor is in the 50 percent range.

(6) The time penalty associated with the lower initial iteration factors is greater for the MQM than the other indirect methods.

(7) The MPF is superior to the MAF and MQM when convergence time is considered, because of the one less equation that must be integrated.

(8) The approximations to the Lagrange multiplier values as derived by the MSD are well within the convergence envelopes of all the indirect method investigated.

CHAPTER 7

DESCRIPTION AND EVALUATION OF NUMERICAL PERFORMANCE

The evaluation of numerical performance is an essential feature in assessing the accuracy of an optimization technique. The primary sources of error are encountered during numerical integration and solving of linear systems (which includes matrix inversion). Most of the computational time is taken during numerical integration and hence, increasing the speed of the integration will have a pronounced effect on the total computer time. The criterion used for defining convergence is also a factor in determining total time, and if caution is not exercised an unrealistic comparison between different optimization methods could result.

7.1 Numerical Integration

There are many characteristics that must be considered when selecting a particular numerical integration scheme; some of the most important are accuracy, stability and speed. The method and procedure to be explained takes excellent advantage of the above characteristics.

7.1.1 Numerical Integration Routine

The numerical integration routine consists of two subroutines and either a control subroutine or a control

block of code. A Runge-Kutta fourth-order routine is used as a starter, supplying the initial and three succeeding derivatives. Control is then shifted to a subroutine that contains a fourth-order Adams-Bashford predictor and a fifth-order Adams-Moulton corrector. An option for the iteration of the corrector is provided.

One of the nicest features of the integration package is the method by which the derivatives are stored and moved. The names that refer to these locations are simply changed, rather than changing the location of each derivative itself, and the values are used as if being rolled from a drum. Credit for this unique and time saving idea is given to W. T. Fowler and G. J. Lastman of the Engineering Mechanics Department, The University of Texas.

An additional capability of the subroutine is that the starting value of the integration step size may be subdivided into N substeps, thus providing extremely accurate starting values for the derivatives. The Runge-Kutta is then called 3N times and the derivatives are saved every N^{th} integration step. Four derivatives now being available, the integration proceeds using the usual predict-correct cycle.

7.1.2 Numerical Integration Procedure

The numerical integration proceeds using N = 3 and the Runge-Kutta is called nine times, hence a derivative is

saved on every third substep. This provides the initial four values required by the Adams-Bashford predictor. A constant value of step-size is used to continue the integration.

Two methods are used to terminate the integration, and the method selected depends on whether or not a backwards integration of the adjoint equations is expected. If the adjoint equations are to be integrated, when the remaining time is less than four steps this time is subdivided into 3N substeps and control is shifted to Runge-Kutta. This provides values of the dependent variables which will be used to form coefficients for the backwards integration of the adjoint equations. If backwards integration is not anticipated, when the remaining time is less than one step, control is shifted to Runge-Kutta for the final time increment.

The subdividing of integration steps at the beginning and end of the trajectory increases the programming complexity, however, it was decided that this additional difficulty was more than compensated for by the increase in accuracy of the starting derivatives.

7.1.2.1 Successive Application of Corrector

Successive application of the Adams-Moulton corrector was made for an optimal Earth-Mars transfer trajectory using from one through five applications. No improvement was made

in the optimal values of the Lagrange multipliers and terminal time after the number of applications reached three. Hence, it was decided that two applications of the corrector would be sufficient.

The computation time is reduced by approximately 20 percent when only one application of the corrector is made and increased by approximately 20 percent when three corrections are made.

The selection of a corrector with two iterations was encouraged further by examination of the terminal values of the state variables after the first iteration.

7.1.2.2 Step Size Selection

The step-size of the numerical integration technique is extremely important. Not only does the accuracy of the method depend on this selection, but the resulting computer time as well. So much depends on this selection that a considerable effort for its determination is justified. One complicating factor that exists for comparison studies is that convergence time is to be compared for all methods, some of which might require different integration step sizes.

The criteria that is used in selecting step-size is determined in the following manner:

(1) Use the near optimal starting conditions of -10, -10, and 20 percent error in the initial

Lagrange multipliers and terminal time, respectively. Proceed to a convergent condition using integration step sizes that range on either side of some reasonable value.

(2) Record the resulting optimal values of the Lagrange multipliers and terminal time and the time required for convergence.

(3) Small integration steps result in large round-off errors and large steps result in large truncation errors. A step-size value in the range where a maximum number of signficant figures agree is interpreted as a desirable one.

The integration step-size of 0.03 units of time was chosen for the Earth-Mars transfer because the value of the estimated variables on either side of the selected step agreed to at least five places. The step-size for the Earth launch trajectory was selected to be 2.0 seconds.

The plot in Figure 45 of convergence time as a function of integration step-size for the MAF, MPF, and MQM and the Earth-Mars transfer reveals that a larger step would result in fewer places of numerical agreement, while a smaller step would suffer from a severe time penalty as well as fewer places of agreement.

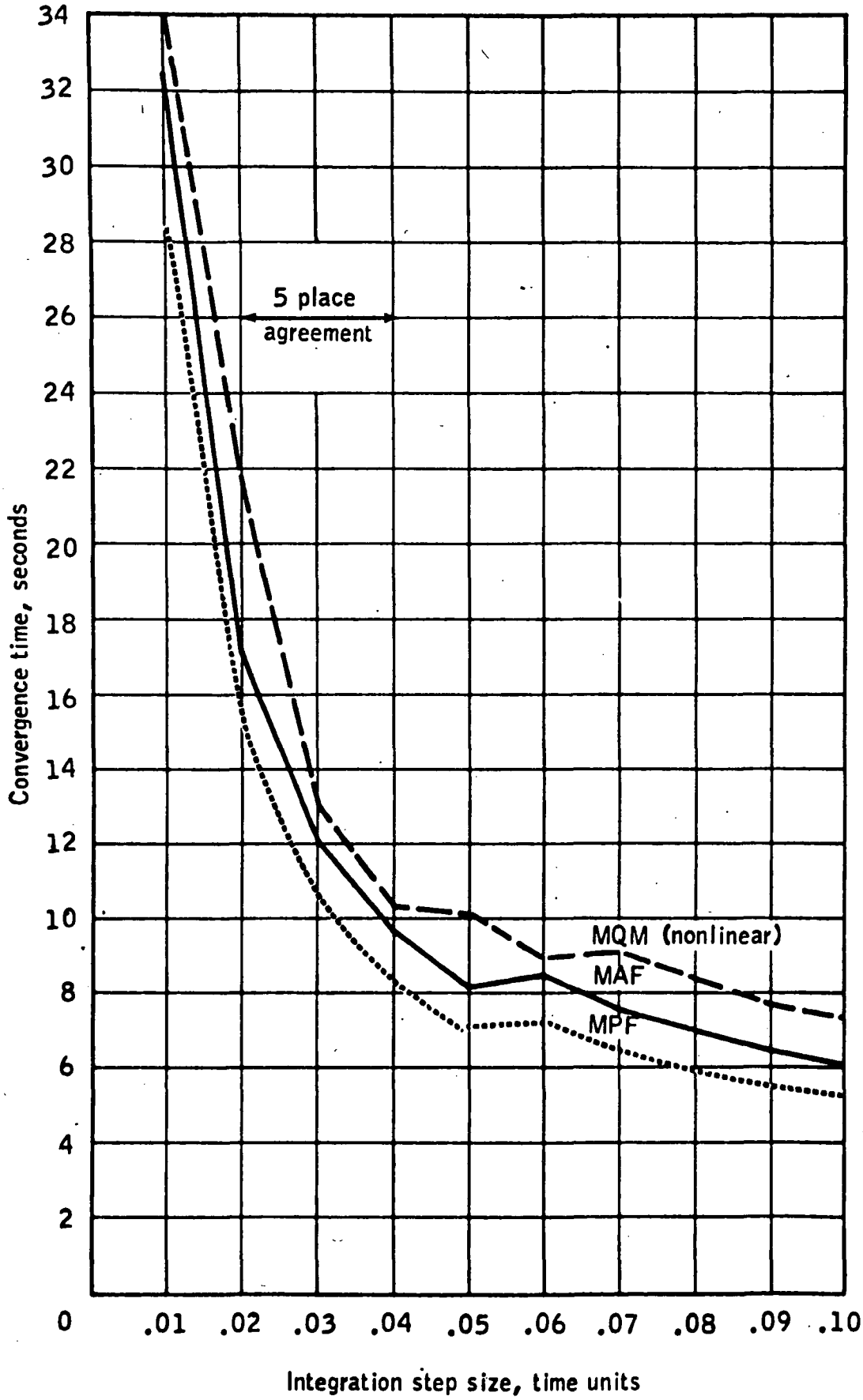


Figure 45. - Convergence time as a function of integration step size using normal iteration scheme.

7.2 Linear System Routine

The computer routine that solves a general linear system of equations $AX = B$ is composed of six subroutines. The routine has the additional capability of returning the determinate of A, an inverse of A, an indication if A is singular and an estimate of the condition number of A.

The first operation of the master driver program is to row equilibrate the matrix A by an exponent procedure. The equilibrating multipliers are stored for later use to scale the right hand side B. An initial estimate of X is determined and a residual vector is found that defines a new linear system. This system is solved and a correction is added to the previous solution. Sufficient information is then available to initiate an iteration for the final solution of X.

7.3 Numerical Criteria Affecting Accuracy

The numerical accuracy of a computer solution depends not only on programming skills, but other criteria as well. For instance, it is desirable in numerical studies to achieve some degree of numerical magnitude compatibility. This is conveniently accomplished by normalizing of the state variables, Lagrange multipliers, and time.

One additional item that affects numerical accuracy is the criterion for establishing when convergence has

occurred. Since it is desired to compare the results of several different optimization methods on a convergence time basis, it is essential that the methods result in the same order of numerical accuracy.

7.3.1 Normalization of Numerical Parameters

In many cases, such as the ones presented here, the correction to several of the variables is used to determine some of the procedures followed in the iteration scheme, even though these variables have different units. Hence, it is desirable, from a computational point of view, to achieve some degree of numerical magnitude compatibility.

This normalization is accomplished for the state variables by selecting certain quantities to be new units of that variable. As shown in Appendix A.4, three variables are selected and these selections dictate new units for the remaining variables. An effort is made to choose the three variables such that the range of all variables is near unity. In an effort to make the Lagrange multipliers numerically compatible with these state variables, a scaling process is used.

In any two-point boundary value problem where $2n$ differential equations are involved, $2n+2$ boundary conditions must be specified, all of which are not necessarily at the same boundary. If an additional initial boundary

condition is obtained, a terminal boundary condition must be ignored. Now, since the Euler Lagrange equations are linear and homogeneous, the solution is simply a linear magnification of the initial conditions.

In the optimization problem, the Lagrange multipliers may be normalized by selecting one multiplier to be positive or negative unity and in this manner adding one initial boundary condition. This simply scales the multipliers by the unnormalized value of this multiplier. With the addition of this initial boundary condition, a terminal condition must be ignored. It is recommended that the ignored terminal condition be one of the conditions that result from the transversality equations because usually there is little intuitive feel for the physical significance of these equations. In requesting a desired improvement in the satisfaction of terminal constraints, it may be helpful to have a intuitive feel for the meaning of these constraints.

The fact that one of the transversality conditions is ignored does not mean that this condition is not satisfied. For instance, if the ignored transversality terminal constraint

$$h = (\phi_x + \lambda^T)_f$$

is perturbed so that the terminal dissatisfaction becomes

$$dh = (\phi_{xx}dx + \phi_{xt}dt + d\lambda^T)_f$$

it is seen that when the solution does converge, the terminal dissatisfaction vanishes because dx_f, dt_f, and $d\lambda_f^T$ vanish.

7.3.2 Criteria for Defining Convergence

Establishing when convergence has occurred is an essential part of determining the characteristics of a convergence process. Defining convergence becomes a matter of arbitration.

In the present study the criterion used is that the corrections being applied to the initial estimates of the Lagrange multipliers and terminal time must be less than some small number. There are, however, several other trajectory characteristics that must be observed. For instance, in the MAF and MPF an improvement in the terminal constraints is requested, but this request is not always completely effective. Therefore, the norm of the terminal constraints is improved as the method proceeds, and hence the convergence definition could hinge on the terminal dissatisfaction being less than some small number. Even if this criterion is not used, as in the case presented here, the norm of terminal dissatisfaction is of great interest and should be observed closely.

In the investigation of the MGNR the terminal constraints are satisfied identically, but the trajectory shape does not correspond to the shape assumed by a trajectory that satisfies the optimality conditions. Hence, one logical criterion for this method is a metric that represents the maximum distance between corresponding time points on the present and previous trajectory. This metric is recorded and is used in the selection of the correction criterion.

The iteration procedure for the indirect methods continue until change in the norm of terminal dissatisfaction between the final two iterations in MAF and MPF is comparable in numerical magnitude to the metric described in MGNR. These criteria for establishing convergence may result in slightly different values of correction criterion for the different methods. The over-riding factor of concern is that trajectories to be compared should have approximately the same numerical accuracy.

A correction criterion of 10^{-6} for an Earth-Mars transfer using MAF and MPF produced a final terminal norm change of order 10^{-5}. The correction criterion that resulted in a metric of approximately 10^{-5} was also 10^{-6}. The MSD is difficult to compare with the indirect methods since convergence in the same sense is never reached.

7.4 Computation Facilities

The numerical investigation was made at the facilities of NASA-Manned Spacecraft Center, Houston, Texas. The facility used for the numerical calculations was the directly coupled IBM 7094. All programs were programmed in FORTRAN IV compiler language.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

There have been many significant conclusions based on both the theoretical and numerical results described in the previous chapters. Detailed results and conclusions have been presented in Sections 6.3, 6.4, and 6.5. In Section 6.6, a summary of the relative merits of the methods is made with particular emphasis on the basis of comparison as explained in Section 6.2. A general summary of the most significant conclusions are presented in this chapter.

The many questions that have been successfully answered during this investigation have brought forth many new unanswered questions, and this is as it should be. The existence of these new questions provide a motivation for additional and perhaps rewarding studies, and several possibilities for continued investigation are suggested.

8.1 Summary of Conclusions

The major theoretical conclusions resulting from the analysis are:

(1) The Method of Adjoint Functions and the Method of Perturbation Functions are recognized as essentially the same method. The Method of Perturbation Functions, however, requires one less integration

because of the more efficient manner in which the coefficient matrix of the perturbation equation is generated.

(2) The Modified Quasilinearization Method is an extension of the Method of Generalized Newton-Raphson which accommodates problems that have terminal boundaries given as general functions of the state and/or Euler variables. Moreover, the terminal time determination is made an integral part of the iterative procedure itself, and no additional terms must be added to the existing differential equations and no additional differential equations are needed.

(3) A unique and easily determined weighting matrix has been derived which increases the convergence rate of the Method of Steepest Descent. This matrix assists the method in accelerating the shaping of the optimal control program during the terminal iterations.

The other major conclusions resulting from the analysis are:

(1) Two iteration schemes which significantly increase the possibility for convergence have been successfully implemented for the indirect methods. This desirable characteristic is obtained with one of the schemes without an appreciable increase in convergence time.

(2) The Modified Quasilinearization Method is successfully implemented and results in a significant decrease in convergence time when compared to the other quasilinearization methods studied.

(3) The Method of Steepest Descent, after only a few iterations, provides initial values of the Lagrange multipliers which are well within the convergence envelopes of all the indirect methods investigated.

The results of this investigation support the claim that a hybrid optimization method would be the most desirable method to build for a general purpose capability. This hybrid method would consist of the Method of Steepest Descent for the initial phase of optimization and switch to the Method of Perturbation Functions for the later phase. It must be pointed out, however, that building a general purpose optimization method would result in a very time consuming method, whereas by knowing the specific nature of a given situation, a very efficient method can be tailor made for that particular situation.

8.2 Recommendations for Continued Study

The present investigation has succeeded in developing a new method, based on the theory of quasilinearization, which places the Quasilinearization Methods in a more competitive position with the Perturbation and Gradient Methods. Several iteration schemes are formulated and applied, and significant

reductions in computation time and initial parameter sensitivity have been realized. A foundation has been laid for building more complex methods which will in turn handle more complex and realistic problems.

A natural extension of the current investigation would be to study several more example problems that have a larger dimension, more control variables and that require inequality constraints, such as a three-dimensional, atmospheric, reentry problem.

Some thought has been given to developing a method for approximating the initial values of the Lagrange multipliers by assuming a control program for the first iteration in the indirect methods, or by using the constants of motion as derived by Melbourne (28).

The applicability of several other methods for solving the nonlinear two-point boundary value problem, associated with the trajectory optimization problem, should be investigated, such as the ones proposed by Merriam (29) and Sylvester and Meyer (30). A comparison should be made between the methods discussed in this study and the methods recently proposed by McReynolds and Bryson (24) and Kopp and Moyer (11).

A generalized hybrid optimization program may be easily built in which the initial values of the Lagrange multipliers are approximated by using a direct method, then switching, when the estimates are within the convergence envelope, to an indirect method for rapid convergence. The details of such a procedure should be studied.

Application of the Reduction of an Optimization Problem to a Two-Point Boundary Value Problem

The following application is formulated to illustrate the procedure explained in Section 2.2. The equation numbers in parenthesis refer to the corresponding equation in Section 2.2. The nonlinear, ordinary, differential equations of motion are

$$\begin{aligned}
\dot{x}_1 &= \dot{u} = \frac{v^2}{r} - \frac{GM}{r^2} + \frac{T \sin\beta}{m} = f_1 \\
\dot{x}_2 &= \dot{v} = -\frac{uv}{r} + \frac{T \cos\beta}{m} = f_2 \\
\dot{x}_3 &= \dot{r} = u = f_3 \\
\dot{x}_4 &= \dot{\theta} = \frac{v}{r} = f_4
\end{aligned} \qquad \begin{matrix}(A.1.1)\\(2.23)\end{matrix}$$

and the nonlinear, ordinary, Euler-Lagrange differential equations are

$$\begin{aligned}
\dot{\lambda}_1 &= \left(\frac{v}{r}\right)\lambda_2 - \lambda_3 = f_5 \\
\dot{\lambda}_2 &= -\left(\frac{2v}{r}\right)\lambda_1 + \left(\frac{u}{r}\right)\lambda_2 - \left(\frac{1}{r}\right)\lambda_4 = f_6 \\
\dot{\lambda}_3 &= \left(\frac{v^2}{r^2} - \frac{2GM}{r^3}\right)\lambda_1 - \left(\frac{uv}{r^2}\right)\lambda_2 + \left(\frac{v}{r^2}\right)\lambda_4 = f_7 \\
\dot{\lambda}_4 &= 0 = f_8
\end{aligned} \qquad \begin{matrix}(A.1.2)\\(2.24)\end{matrix}$$

The optimality condition $H_u = 0$ leads to

$$\frac{T}{m}(\lambda_1 \cos \beta - \lambda_2 \sin \beta) = 0. \qquad \begin{matrix}(A.1.3)\\(2.25)\end{matrix}$$

This condition implies that

$$\tan \beta = \frac{\lambda_1}{\lambda_2} \qquad \sin \beta = \frac{\lambda_1}{\pm\sqrt{\lambda_1^2+\lambda_2^2}} \qquad \cos \beta = \frac{\lambda_2}{\pm\sqrt{\lambda_1^2+\lambda_2^2}}$$

where the sign in front of the radical terms is selected according to the Weierstrass E-Function.

The Weierstrass Condition is the fourth necessary condition which must be satisfied for a given trajectory to be an extremal. It is defined as

$$E=F(x^*,\dot{x},t)-F(x^*,\dot{x}^*,t)-\frac{\partial F}{\partial \dot{x}^*}(x^*,\dot{x}^*,t)\,(\dot{x}-\dot{x}^*) \geq 0 \qquad (A.1.4)$$

for a minimum where E is the Weierstrass E-Function and $F = \lambda^T(f - \dot{x})$. The asterisk refers to the optimal trajectory.

Since the equations of motion must be satisfied on the optimal, as well as the nearby trajectory, $F = F^* = 0$ and the Weierstrass E-Function becomes

$$E = \lambda^T(\dot{x} - \dot{x}^*). \qquad (A.1.5)$$

Making the proper substitutions in Eq. (A.1.5) yields

$$E=\lambda_1\left[\frac{T}{m}(\sin\beta-\sin\beta^*)\right] + \lambda_2\left[\frac{T}{m}(\cos\beta - \cos\beta^*)\right] \geq 0. \qquad (A.1.6)$$

The optimality conditions, i.e. Eq. (A.1.3), leads to the requirement that

$$\tan \beta^* = \frac{\lambda_1}{\lambda_2}$$

which implies

$$\sin \beta^* = \frac{\lambda_1}{\pm\sqrt{\lambda_1^2+\lambda_2^2}} \quad \text{and} \quad \cos \beta^* = \frac{\lambda_2}{\pm\sqrt{\lambda_1^2+\lambda_2^2}} \,. \tag{A.1.7}$$

Eq. (A.1.3) does not indicate which sign should be selected on the radical terms. Substituting Eq. (A.1.7) into Eq. (A.1.6) yields

$$E = \frac{T}{m}\left[\pm\sqrt{\lambda_1^2+\lambda_2^2}\right]\left[-1 + \cos(\beta - \beta^*)\right] \geq 0 \text{ and} \tag{A.1.8}$$

for this equation to be satisfied for all admissible values of β, the negative sign on the radical must be chosen. Hence, the optimal control program is given by

$$\sin \beta^* = \frac{\lambda_1}{-\sqrt{\lambda_1^2+\lambda_2^2}}$$

$$\cos \beta^* = \frac{\lambda_2}{-\sqrt{\lambda_1^2+\lambda_2^2}} \,.$$

The specified initial boundary conditions are

$$\begin{aligned} \eta_1 &= g_1 = u(t_0) - u_0 = 0 \\ \eta_2 &= g_2 = v(t_0) - v_0 = 0 \\ \eta_3 &= g_3 = r(t_0) - r_0 = 0 \\ \eta_4 &= g_4 = \theta(t_0) - \theta_0 = 0 \end{aligned} \tag{A.1.9}$$
$$(2.26)$$

where t_0 is specified. Hence no initial conditions are obtained from the transversality conditions because the initial state and time are specified.

The specified terminal boundary conditions are

$$\Psi_1 = h_1 = u(t_f) - u_f = 0$$

$$\Psi_2 = h_2 = v(t_f) - v_f = 0 \qquad \text{(A.1.10) (2.29)}$$

$$\Psi_3 = h_3 = r(t_f) - r_f = 0 \ .$$

If it is desired to determine the minimum time transfer, the performance index is $\Phi = t_f$, and the terminal transversality conditions are

$$-(\lambda_1 du + \lambda_2 dv + \lambda_3 dr + \lambda_4 d\theta)_f + \qquad \text{(A.1.11) (2.35)}$$

$$(1 + \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4)_f \, dt_f = 0 \ .$$

The terminal state perturbations in Eq. (A.1.11) are not independent. They are related through Eq. (2.36). The application of this equation results in

$$du_f = dv_f = dr_f = 0 \ . \qquad \text{(A.1.12)}$$

Combining Eqs. (A.1.11) and (A.1.12), the fourth terminal boundary condition becomes

$$h_4 = \lambda_{4f} = 0 \qquad \text{(A.1.13)}$$

since it is not desired to constrain the terminal value of the angle θ. If, however, it is desired to constrain the terminal angle, $d\theta_f$ must vanish and λ_{4f} would not

necessarily be zero. In this case, the fourth terminal boundary condition becomes

$$h_4 = \theta(t_f) - \theta_f = 0 \ . \qquad \text{(A.1.14)}$$

Allowing for the possibility of a variable terminal time, Eq. (A.1.11) also yields the fifth and last terminal condition

$$h_5 = (1 + \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4)_f = 0. \qquad \text{(A.1.15)}$$

If it is desired to normalize the Lagrange multipliers as discussed in Section 7.3.1, one multiplier is initially selected plus or minus unity and one terminal boundary condition is ignored. The initial boundary condition $\lambda_3 (t_0) = -1.0$, is used in place of the fifth terminal boundary condition, and the result is

$$\begin{aligned} g_1 &= u(t_0) - u_0 = 0 & h_1 &= u(t_f) - u_f = 0 \\ g_2 &= v(t_0) - v_0 = 0 & h_2 &= v(t_f) - v_f = 0 \\ g_3 &= r(t_0) - r_0 = 0 & h_3 &= r(t_f) - r_f = 0 \\ g_4 &= \theta(t_0) - \theta_0 = 0 & h_4 &= \lambda_4(t_f) = 0 \\ g_5 &= \lambda_3(t_0) + 1.0 = 0 \end{aligned} \qquad \text{(A.1.16)}$$

For the solution of $2n$ differential equations, $2n+2$ boundary conditions must be known. Assuming that the initial time is zero, $2n+1$ conditions are needed. These are the boundary conditions given above.

Discussion of the Applications

The example class of problems used to apply the theoretical formulations presented in Chapters 3, 4, and 5 is the minimum time trajectory of a thrusting spacecraft under the influence of an inverse square gravitational force field. The specific examples used to obtain the numerical results discussed in Chapter 6 are:

(1) A constant low thrust Earth-Mars transfer trajectory which leaves the Earth's circular orbit about the Sun with a velocity equal to that of the Earth. The control or thrust angle is unbounded and only the Sun's gravitational influence is considered. The spacecraft arrives at an arbitrary heliocentric angle in the circular Mars orbit having velocity conditions that match that of Mars.

(2) A constant high thrust Earth launch to a 100 kilometer circular orbit leaving the Earth's surface with zero velocity. The control or thrust angle is unbounded. The Earth's inverse square gravitational influence is considered. The dissapative terms of the atmospheric drag are also included. The spacecraft arrives at an arbitrary heliocentric

angle in the circular orbit. The effects of other bodies are neglected.

In the optimization reduction problem shown in Appendix A.1 it is seen that the initial state is specified and hence $n = p = 4$. The terminal velocity and radial position are specified and hence $q = 3$. Two additional terminal constraints are derived from the transversality conditions. Assuming that the initial time is specified as zero, five initial conditions and five terminal conditions are specified, therefore the problem is solvable.

When the numerical parameters are normalized as discussed in Section 7.3.1, the initial value of the Lagrange multiplier associated with the radius is equated to a negative unity, and hence $p = 5$, and the last transversality condition is ignored. This means that six initial conditions and four terminal conditions are specified, where the initial time is included. The problem is still solvable, but the complexion of the applications is changed slightly from that described in the detailed procedures presented in Chapters 3 and 4.

It should be pointed out that the fourth differential equation of state and the corresponding Euler-Lagrange equation is not necessary for the analysis made here. These equations are simply included for the sake of generality,

and hence the same computer programs may easily be converted to solve the class of problems where terminal state is completely specified.

The time histories of each variable that correspond to the optimal solution for the Earth-Mars transfer are illustrated in Figure A.5.1. The optimal control history for this problem is shown in Figure A.5.2. The time histories of each variable that correspond to the optimal solution for the Earth launch are illustrated in Figure A.5.3. The optimal control history for this problem is shown in Figure A.5.4.

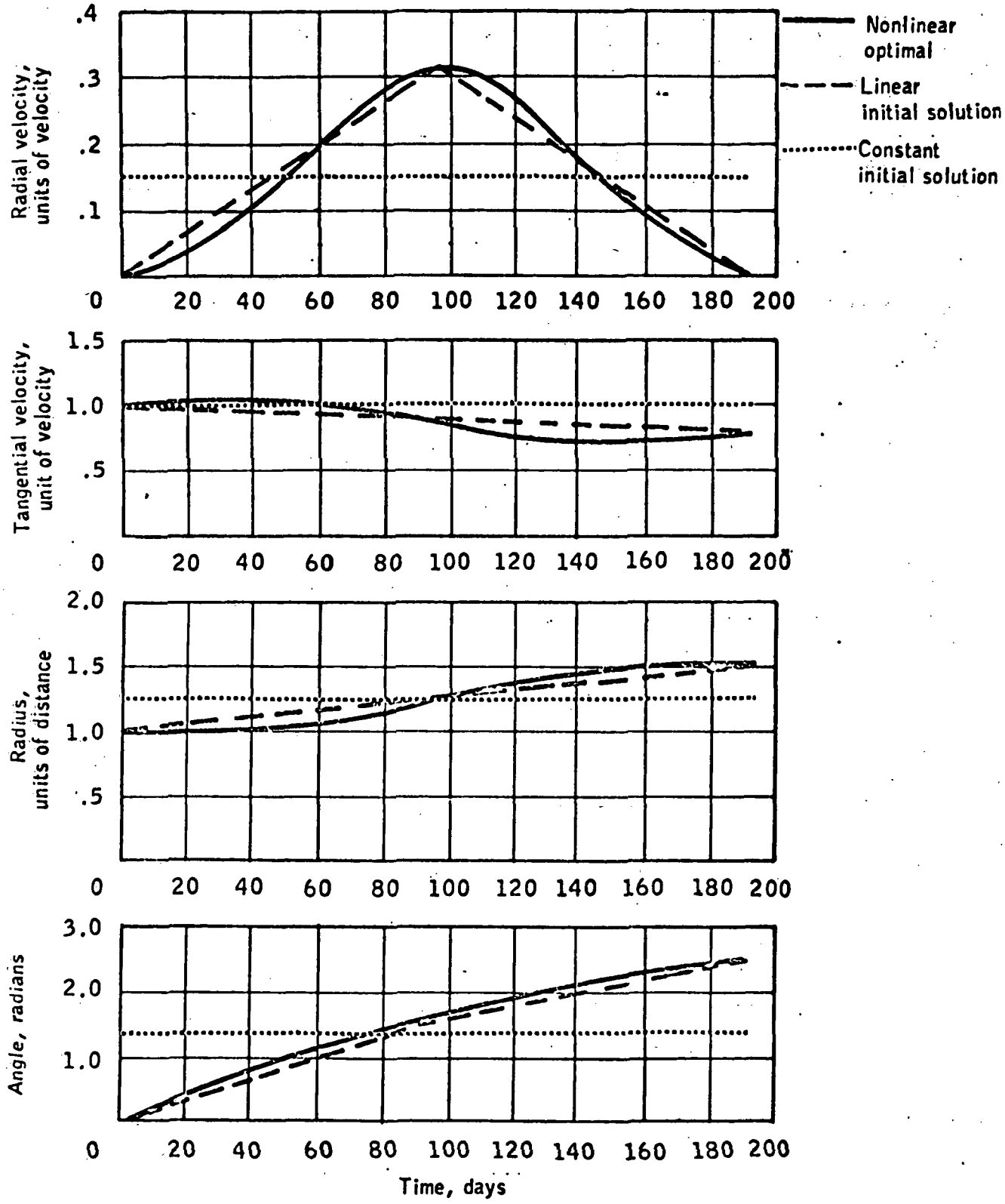
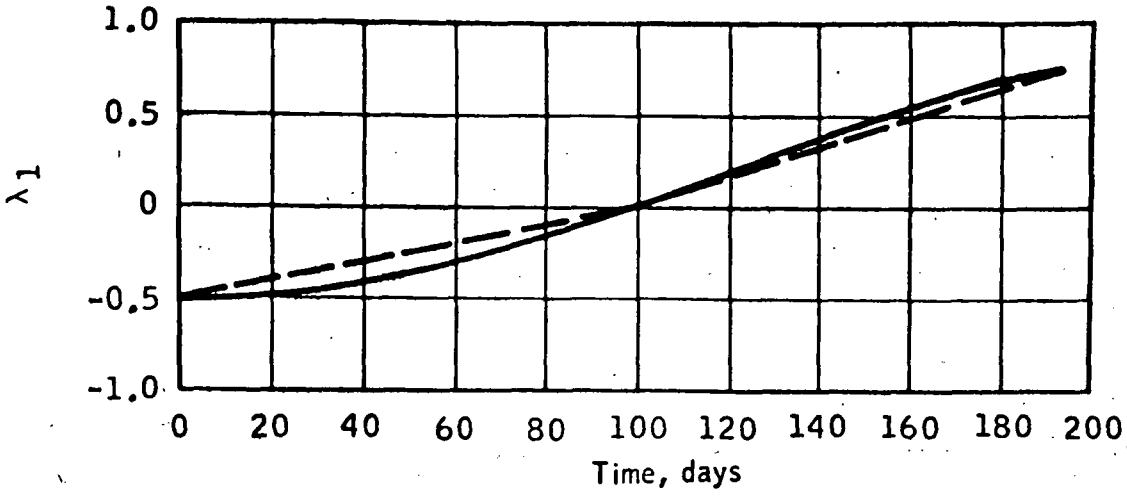
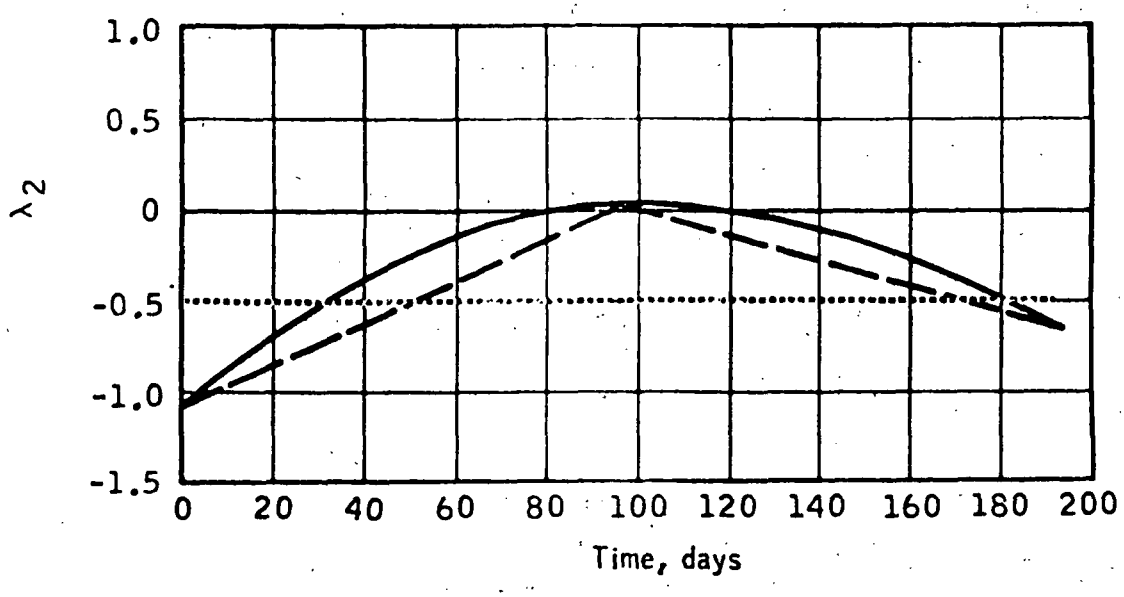


Figure A.2.1 - Optimal trajectory for the Earth-Mars transfer.





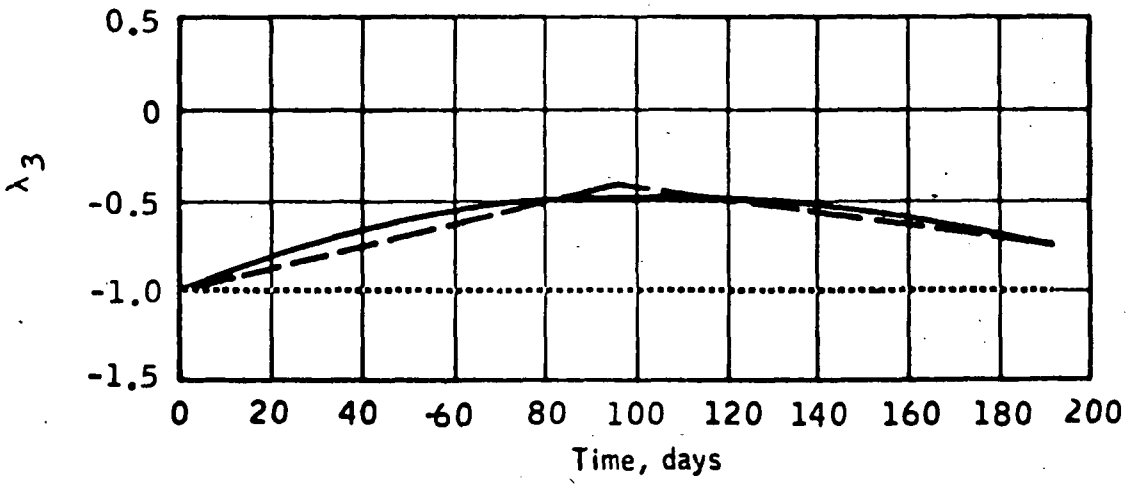


Figure A.2.1. - Optimal trajectory for Earth-Mars transfer (con't).

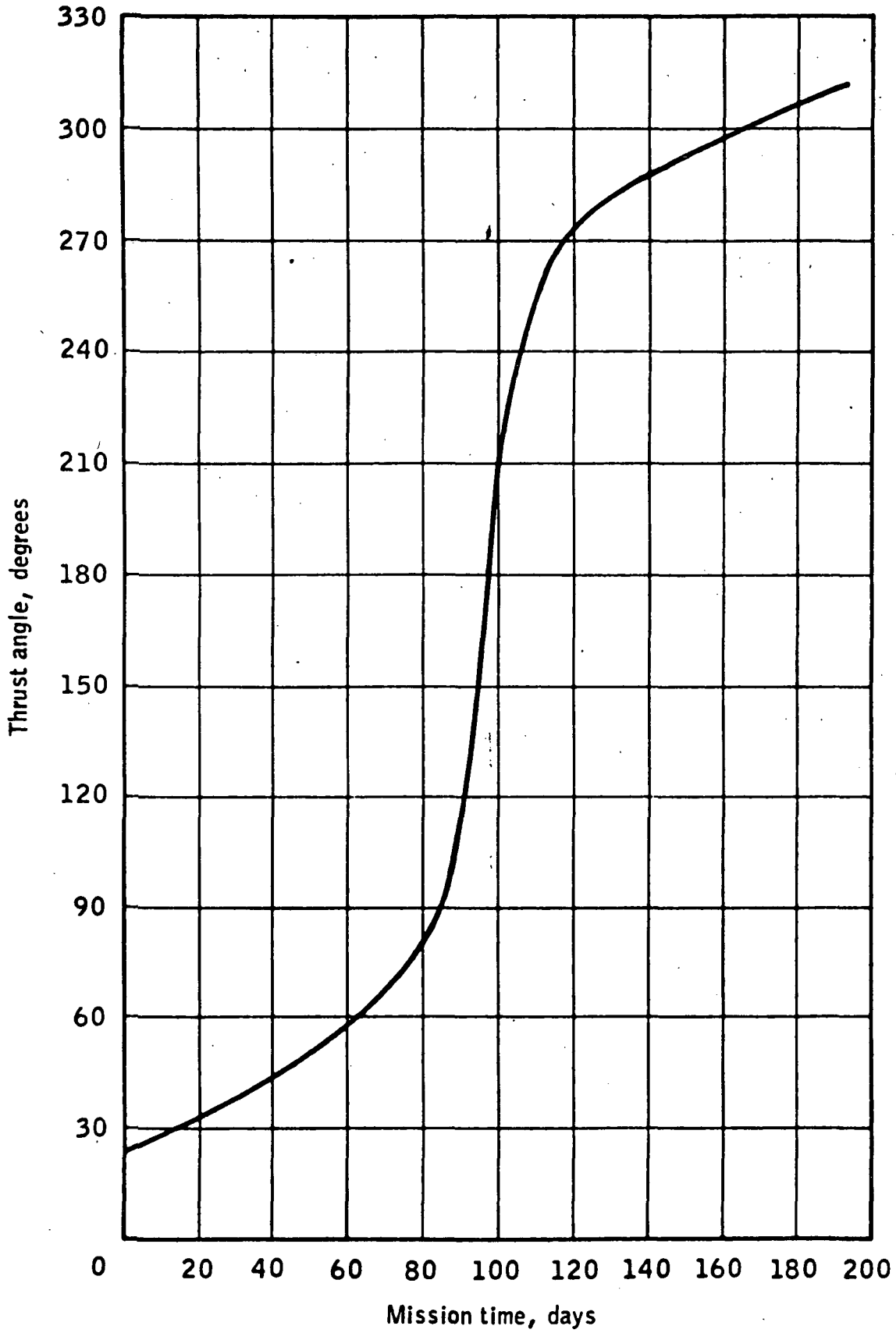


Figure A.2.2. - Optimal constant and unbounded thrust program for the Earth-Mars transfer.

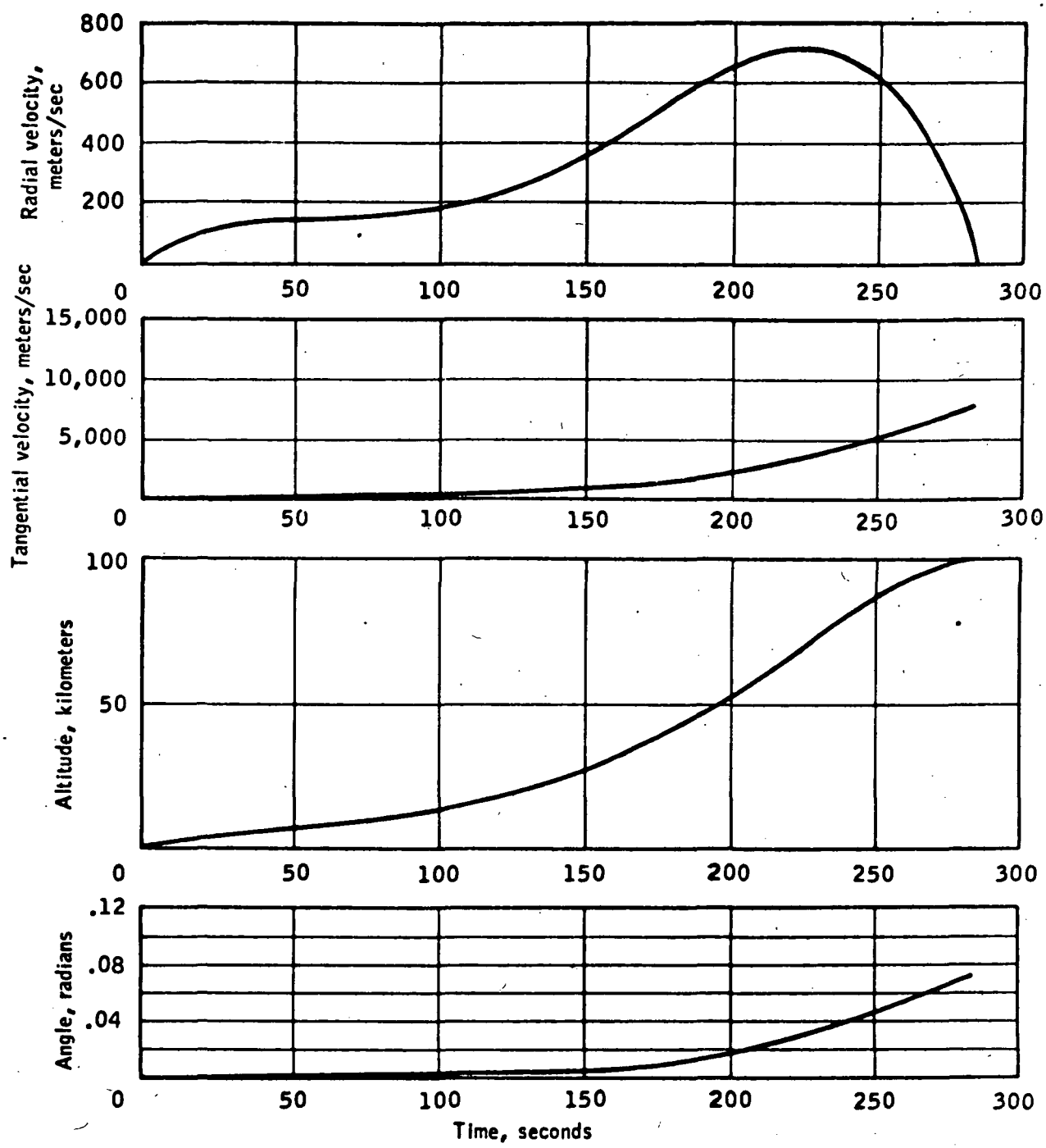


Figure A.2.3. - Optimal trajectory for the atmospheric Earth launch.

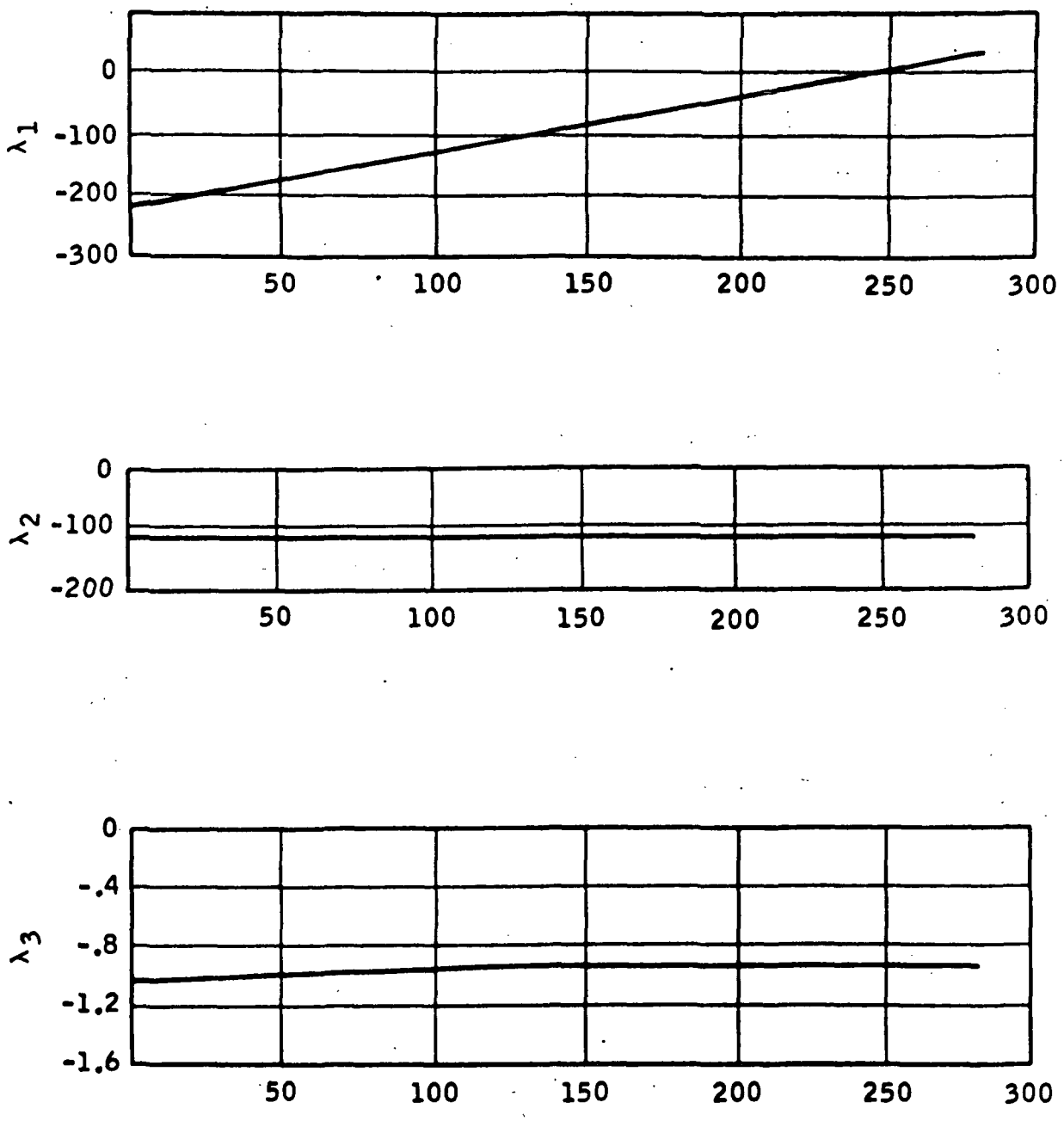


Figure A.2.3. - Optimal trajectory for the atmospheric Earth launch (cont).

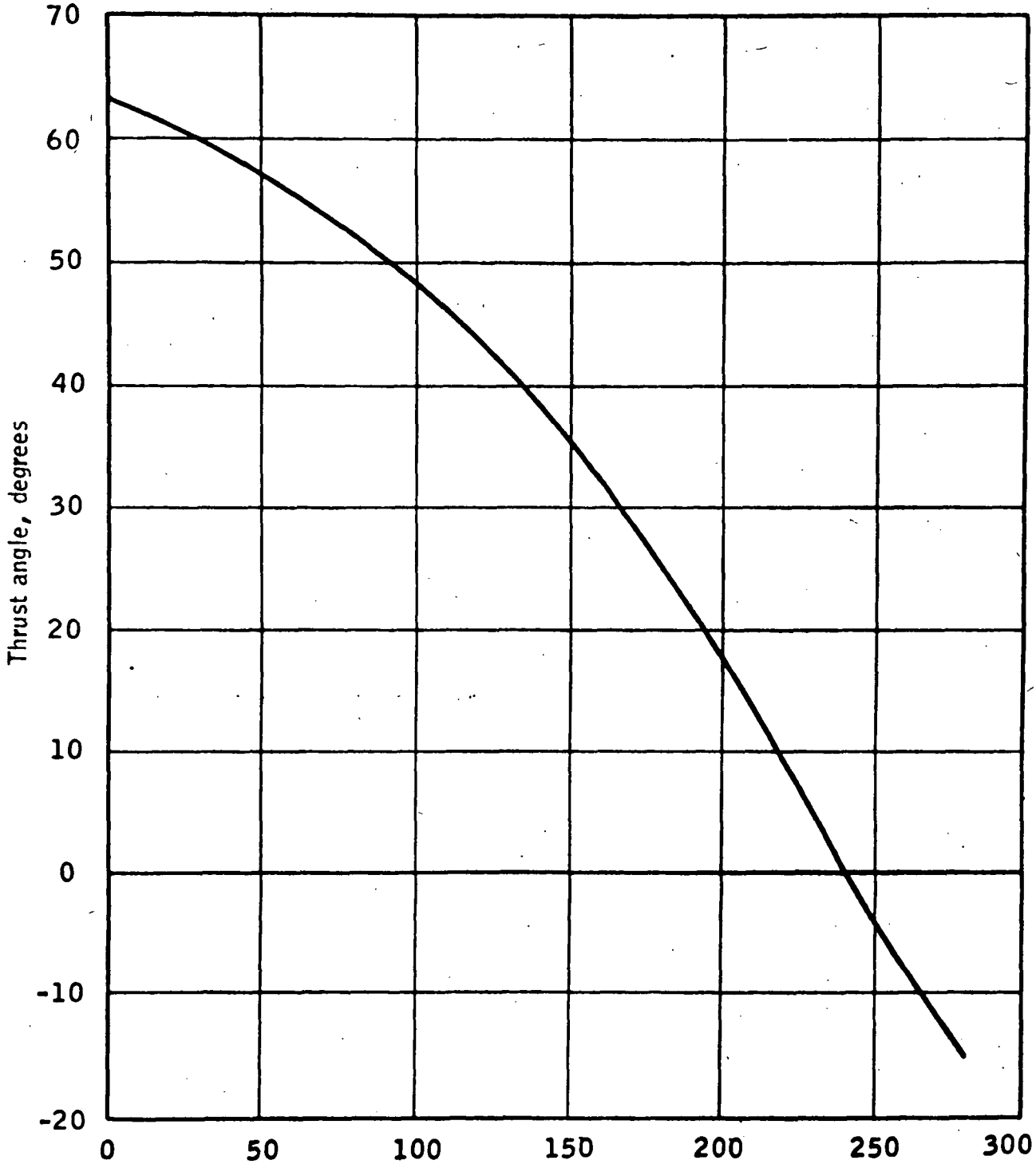


Figure A.2.4. - Optimal constant and unbounded thrust program for the atmospheric Earth launch.

APPENDIX A.2.1

Application of the Method of Adjoint Functions

The nonlinear, ordinary, vector differential equation $\dot{z} = F(z,t)$ is composed of $n = 4$ differential equations of motion (with control eliminated by use of the optimality condition) and $n = 4$ Euler-Lagrange equations. These equations are integrated from a known t_0 to an assumed $\bar{t}_f$ with the known initial conditions and assumed values for those not known, i.e.

$$z(t_0) = \begin{bmatrix} u \\ v \\ r \\ \theta \\ \bar{\lambda}_1 \\ \bar{\lambda}_2 \\ \lambda_3 \\ \bar{\lambda}_4 \end{bmatrix}_0$$

where the bar indicates an assumed value.

When the assumed terminal time $\bar{t}_f$ is reached, the terminal dissatisfaction h and dissatisfaction rate $\dot{h}$ are evaluated. The starting vectors for the backwards integration of the adjoint equations are also evaluated.

These adjoint equations, $\dot{\Lambda} = -\left[\frac{\partial F}{\partial z}\right]^T \Lambda$, are

$$\dot{\Lambda}_1 = \left(\frac{v}{r}\right)\Lambda_2 - \Lambda_3 - \left(\frac{\lambda_2}{r}\right)\Lambda_6 + \left(\frac{v\lambda_2}{r^2}\right)\Lambda_7$$

$$\dot{\Lambda}_2 = -\left(\frac{2v}{r}\right)\Lambda_1 + \left(\frac{u}{r}\right)\Lambda_2 - \left(\frac{1}{r}\right)\Lambda_4 - \left(\frac{\lambda_2}{r^2}\right)\Lambda_5 + \left(\frac{2\lambda_1}{r}\right)\Lambda_6 - \left[\frac{1}{r^2}\left(2v\lambda_1 - u\lambda_2 + \lambda_4\right)\right]\Lambda_7$$

$$\dot{\Lambda}_3 = \left(\frac{v^2}{r^2} - \frac{2GM}{r^3}\right)\Lambda_1 - \left(\frac{uv}{r^2}\right)\Lambda_2 + \left(\frac{v}{r^2}\right)\Lambda_4 + \left(\frac{v\lambda_2}{r^2}\right)\Lambda_5 - \left[\frac{1}{r^2}\left(2v\lambda_1 - u\lambda_2 + \lambda_4\right)\right]\Lambda_6 - \left\{\frac{2}{r^3}\left[uv\lambda_2 - v\lambda_4 - v^2\lambda_1\right] + \frac{1}{r^4}\left[6GM\lambda_1\right]\right\}\Lambda_7$$

$$\dot{\Lambda}_4 = 0$$

$$\dot{\Lambda}_5 = \left[\frac{T\lambda_2^{\,2}}{m(\lambda_1^{\,2}+\lambda_2^{\,2})^{3/2}}\right]\Lambda_1 - \left[\frac{T\lambda_1\lambda_2}{m(\lambda_1^{\,2}+\lambda_2^{\,2})^{3/2}}\right]\Lambda_2 + \left(\frac{2v}{r}\right)\Lambda_6 - \left(\frac{v^2}{r^2} - \frac{2GM}{r^3}\right)\Lambda_7$$

$$\dot{\Lambda}_6 = -\left[\frac{T\lambda_1\lambda_2}{m(\lambda_1^{\,2}+\lambda_2^{\,2})^{3/2}}\right]\Lambda_1 + \left[\frac{T\lambda_1^{\,2}}{m(\lambda_1^{\,2}+\lambda_2^{\,2})^{3/2}}\right]\Lambda_2 - \left(\frac{v}{r}\right)\Lambda_5 - \left(\frac{u}{r}\right)\Lambda_6 + \left(\frac{uv}{r^2}\right)\Lambda_7$$

$$\dot{\Lambda}_7 = \Lambda_5$$

$$\dot{\Lambda}_8 = \left(\frac{1}{r}\right)\Lambda_6 - \left(\frac{v}{r^2}\right)\Lambda_7 ,$$

and are integrated backwards from t_f to t_0 forming the coefficients from the variables stored during the forward

integration. The $2n+1-p = 4$ starting vectors for this backward integration are

$$\Lambda_1(t_f) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Lambda_2(t_f) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Lambda_3(t_f) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \Lambda_4(t_f) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

When the initial time t_0 is reached, $2n+1-p$ algebraic equations are solved for the linear estimates for the corrections that must be applied to the assumed initial conditions $(\bar{\lambda}_{10}, \bar{\lambda}_{20}, \bar{\lambda}_{40})$ and the assumed terminal time $(\bar{t}_f)$. These algebraic equations are

$$\begin{bmatrix} \delta\lambda_1(t_0) \\ \delta\lambda_2(t_0) \\ \delta\lambda_4(t_0) \\ \delta t_f \end{bmatrix} = \begin{bmatrix} \theta_{51} & \theta_{61} & \theta_{81} & \dot{u}_f \\ \theta_{52} & \theta_{62} & \theta_{82} & \dot{v}_f \\ \theta_{53} & \theta_{63} & \theta_{83} & \dot{r}_f \\ \theta_{54} & \theta_{64} & \theta_{84} & \lambda_{4f} \end{bmatrix}^{-1} \begin{bmatrix} du \\ dv \\ dr \\ d\lambda_4 \end{bmatrix}_f$$

where the elements of the θ matrix are evaluated at t_0. These corrections are applied to the initially assumed values of $\bar{\lambda}_1$, $\bar{\lambda}_2$, $\bar{\lambda}_4$ and $\bar{t}_f$ and a new nominal trajectory is integrated using $\dot{z} = F(z,t)$.

APPENDIX A.2.2

Application of the Method of Perturbation Functions

The nonlinear, ordinary, vector differential equation $\dot{z} = F(z,t)$ is composed of $n = 4$ differential equations of motion (with control eliminated by use of the optimality condition) and $n = 4$ Euler-Lagrange equations. These equations are integrated from a known t_0 to an assumed $\bar{t}_f$ with the known initial conditions and assumed values for those not known, i.e.,

$$z(t_0) = \begin{bmatrix} u \\ v \\ r \\ \theta \\ \bar{\lambda}_1 \\ \bar{\lambda}_2 \\ \lambda_3 \\ \bar{\lambda}_4 \end{bmatrix}_0$$

where the bar indicates an assumed value.

The perturbation equations $\delta\dot{z} = \left[\frac{\partial F}{\partial z}\right]\delta z$ are

$$\delta\dot{z}_1 = \left(\frac{2v}{r}\right)\delta z_2 + \left(\frac{2GM}{r^3} - \frac{v^2}{r^2}\right)\delta z_3 - \left[\frac{T\lambda_2{}^2}{m(\lambda_1{}^2+\lambda_2{}^2)^{3/2}}\right]\delta z_5$$

$$+\left[\frac{T\lambda_1\lambda_2}{m(\lambda_1{}^2+\lambda_2{}^2)^{3/2}}\right]\delta z_6$$

$$\dot{\delta z}_2 = -\left(\frac{v}{r}\right)\delta z_1 - \left(\frac{u}{r}\right)\delta z_2 + \left(\frac{uv}{r^2}\right)\delta z_3 + \left[\frac{T\lambda_1\lambda_2}{m(\lambda_1^2+\lambda_2^2)^{3/2}}\right]\delta z_5 - \left[\frac{T\lambda_1^2}{m(\lambda_1^2+\lambda_2^2)^{3/2}}\right]\delta z_6$$

$$\dot{\delta z}_3 = \delta z_1$$

$$\dot{\delta z}_4 = \left(\frac{1}{r}\right)\delta z_2 - \left(\frac{v}{r^2}\right)\delta z_3$$

$$\dot{\delta z}_5 = \left(\frac{\lambda_2}{r}\right)\delta z_2 - \left(\frac{v\lambda_2}{r^2}\right)\delta z_3 + \left(\frac{v}{r}\right)\delta z_6 - \delta z_7$$

$$\dot{\delta z}_6 = \left(\frac{\lambda_2}{r}\right)\delta z_1 - \left(\frac{2\lambda_1}{r}\right)\delta z_2 + \left[\frac{1}{r^2}(2v\lambda_1 - u\lambda_2 + \lambda_4)\right]\delta z_3 - \left(\frac{2v}{r}\right)\delta z_5 + \left(\frac{u}{r}\right)\delta z_6 - \left(\frac{1}{r}\right)\delta z_8$$

$$\dot{\delta z}_7 = -\left(\frac{v\lambda_2}{r^2}\right)\delta z_1 + \left[\frac{1}{r^2}(2v\lambda_1 - u\lambda_2 + \lambda_4)\right]\delta z_2 + \left[\frac{1}{r^3}\left(\frac{6GM\lambda_1}{r} - 2v^2\lambda_1 + 2uv\lambda_2 - 2v\lambda_4\right)\right]\delta z_3 + \left(\frac{v^2}{r^2} - \frac{2GM}{r^3}\right)\delta z_5 - \left(\frac{uv}{r^2}\right)\delta z_6 + \left(\frac{v}{r^2}\right)\delta z_8$$

$$\dot{\delta z}_8 = 0 \ ,$$

and are integrated forwards from t_0 to t_f forming the coefficients from the variables calculated by the integration

of $\dot{z} = F(z,t)$. The integration of the differential and perturbation equations may be done simultaneously, where the $2n-p = 3$ starting vectors for the perturbation equations are

$$\delta z_1(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \delta z_2(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \delta z_3(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

When the terminal time is reached, $2n+1-p = 4$ algebraic equations are solved for the linear estimates for the corrections that must be applied to the assumed initial conditions $(\bar{\lambda}_{10}, \bar{\lambda}_{20}, \bar{\lambda}_{40})$ and the assumed terminal time $(\bar{t}_f)$. These algebraic equations are

$$\begin{bmatrix} \delta\lambda_1(t_0) \\ \delta\lambda_2(t_0) \\ \delta\lambda_4(t_0) \\ dt_f \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} & \dot{u}_f \\ \Phi_{21} & \Phi_{22} & \Phi_{23} & \dot{v}_f \\ \Phi_{31} & \Phi_{32} & \Phi_{33} & \dot{r}_f \\ \Phi_{81} & \Phi_{82} & \Phi_{83} & \dot{\lambda}_{4f} \end{bmatrix}^{-1} \begin{bmatrix} du \\ dv \\ dr \\ d\lambda_4 \end{bmatrix}_f$$

where the elements of the Φ matrix are evaluated at t_f .

These corrections are applied to the initially assumed values of $\bar{\lambda}_1$, $\bar{\lambda}_2$, $\bar{\lambda}_4$ and $\bar{t}_f$ and a new nominal trajectory is integrated using $\dot{z} = F(z,t)$.

APPENDIX A.2.3

Application of the Modified Quasilinearization Method

The nonhomogeneous linear, ordinary vector differential $\dot{z} = Az + B$ is composed of $n = 4$ linearized differential equations of motion (with the control eliminated by use of the optimality condition) and $n = 4$ linearized Euler-Lagrange equations. These equations are

$$\dot{z}_{1_{n+1}} = \dot{u}_{n+1} = \left(\frac{2v}{r}\right)_n v_{n+1} + \left(\frac{2GM}{r^3} - \frac{v^2}{r^2}\right)_n r_{n+1}$$

$$-\left[\frac{T\lambda_2{}^2}{m(\lambda_1{}^2+\lambda_2{}^2)^{3/2}}\right]_n \lambda_{1_{n+1}} + \left[\frac{T\lambda_1\lambda_2}{m(\lambda_1{}^2+\lambda_2{}^2)^{3/2}}\right]_n \lambda_{2_{n+1}}$$

$$+ (B_1)_n$$

$$\dot{z}_{2_{n+1}} = \dot{v}_{n+1} = -\left(\frac{v}{r}\right)_n u_{n+1} - \left(\frac{u}{r}\right)_n v_{n+1} + \left(\frac{uv}{r}\right)_n r_{n+1}$$

$$+\left[\frac{T\lambda_1\lambda_2}{m(\lambda_1{}^2+\lambda_2{}^2)^{3/2}}\right]_n \lambda_{1_{n+1}} - \left[\frac{T\lambda_1{}^2}{m(\lambda_1{}^2+\lambda_2{}^2)^{3/2}}\right]_n \lambda_{2_{n+1}}$$

$$+ (B_2)_n$$

$$\dot{z}_{3_{n+1}} = \dot{r}_{n+1} = u_{n+1} + (B_3)_n$$

$$\dot{z}_{4_{n+1}} = \dot{\theta}_{n+1} = \left(\frac{1}{r}\right)_n v_{n+1} - \left(\frac{v}{r^2}\right)_n r_{n+1} + (B_4)_n$$

$$\dot{z}_{5_{n+1}} = \dot{\lambda}_{1_{n+1}} = \left(\frac{\lambda_2}{r}\right)_n v_{n+1} - \left(\frac{v\lambda_2}{r^2}\right)_n r_{n+1} + \left(\frac{v}{r}\right)_n \lambda_{2_{n+1}} - \lambda_{3_{n+1}}$$

$$+ (B_5)_n$$

$$\dot{z}_{6_{n+1}} = \dot{\lambda}_{2_{n+1}} = \left(\frac{\lambda_2}{r}\right)_n u_{n+1} - \left(\frac{2\lambda_1}{r}\right)_n v_{n+1}$$

$$+ \left[\frac{1}{r^2}(2v\lambda_1 - u\lambda_2 + \lambda_4)\right]_n r_{n+1} - \left(\frac{2v}{r}\right)_n \lambda_{1_{n+1}} + \left(\frac{u}{r}\right)_n \lambda_{2_{n+1}}$$

$$- \left(\frac{1}{r}\right)_n \lambda_{4_{n+1}} + (B_6)_n$$

$$\dot{z}_{7_{n+1}} = \dot{\lambda}_{3_{n+1}} = -\left(\frac{v\lambda_2}{r^2}\right)_n u_{n+1} + \left[\frac{1}{r^2}(2v\lambda_1 - u\lambda_2 + \lambda_4)\right]_n v_{n+1}$$

$$+ \left[\frac{1}{r^3}\left(\frac{6GM\lambda_1}{r} - 2v^2\lambda_1 + 2uv\lambda_2 - 2v\lambda_4\right)\right]_n r_{n+1}$$

$$+ \left(\frac{v^2}{r^2} - \frac{2GM}{r^3}\right)_n \lambda_{1_{n+1}} - \left(\frac{uv}{r^2}\right)_n \lambda_{2_{n+1}} + \left(\frac{v}{r^2}\right)_n \lambda_{4_{n+1}}$$

$$+ (B_7)_n$$

$$\dot{z}_{8_{n+1}} = \dot{\lambda}_{4_{n+1}} = (B_8)_n$$

where

$$(B_1)_n = \left(-\frac{3GM}{r^2} - \frac{T}{m\sqrt{\lambda_1^2+\lambda_2^2}}\right)_n$$

$$(B_2)_n = -\left(\frac{T\lambda_2}{m\sqrt{\lambda_1^2+\lambda_2^2}}\right)_n$$

$$(B_3)_n = 0$$

$$(B_4)_n = -\left(\frac{v}{r}\right)_n$$

$$(B_5)_n = 0$$

$$(B_6)_n = -\left(\frac{\lambda_4}{r}\right)_n$$

$$(B_7)_n = \left(-\frac{6GM\lambda_1}{r^3} + \frac{v\lambda_4}{r^2}\right)_n$$

$$(B_8)_n = 0 \quad .$$

These nonhomogeneous linear equations are integrated from t_0 to $\bar{t}_f$ with the starting vector

$$z(t_0)_{n+1} = \begin{bmatrix} u \\ v \\ r \\ \theta \\ \bar{\lambda}_1 \\ \bar{\lambda}_2 \\ \lambda_3 \\ \bar{\lambda}_4 \end{bmatrix}_0$$

where the bar indicates an assumed value. This determines the variables for the $n+1^{th}$ iteration by using the variables resulting from the n^{th} iteration to form the required coefficients.

The homogeneous linear equations (same as above except without the $(B_i)_n$, $i = 1, 2n$ terms) $\dot{y} = Ay$, are

integrated from t_0 to t_f in the same manner as the non-homogeneous equations but with the $2n-p=3$ starting vectors

$$y_1(t_0)_{n+1} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad y_2(t_0)_{n+1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad y_3(t_0)_{n+1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}_0 .$$

When the terminal time is reached, $2n+1-p = 4$ algebraic equations are solved for the corrections that must be applied to the assumed initial conditions $(\bar{\lambda}_{10}, \bar{\lambda}_{20}, \bar{\lambda}_{40})$ and the assumed terminal time $(\bar{t}_f)$. These algebraic equations are

$$\begin{bmatrix} \delta\lambda_1(t_0) \\ \delta\lambda_2(t_0) \\ \delta\lambda_4(t_0) \\ dt_f \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & \dot{u}_f \\ y_{21} & y_{22} & y_{23} & \dot{v}_f \\ y_{31} & y_{32} & y_{33} & \dot{r}_f \\ y_{81} & y_{82} & y_{83} & \dot{\lambda}_{4f} \end{bmatrix}^{-1} \begin{bmatrix} du \\ dv \\ dr \\ d\lambda_4 \end{bmatrix}_f$$

where the elements of the matrix are evaluated at t_f.

These corrections are applied to the initially assumed values of $\bar{\lambda}_1$, $\bar{\lambda}_2$, $\bar{\lambda}_4$ and t_f and a new nominal trajectory is integrated using $\dot{z} = Az+B$. where the A and B matrices are formed from the previous nominal.

APPENDIX A.2.4

Application of the Method of Steepest Descent

The nonlinear, ordinary, vector differential equation $\dot{x} = f(x,u,t)$ is composed of $n = 4$ differential equations of motion. These equations are integrated forward with the initial conditions

$$x(t_0) = \begin{bmatrix} u \\ v \\ r \\ \theta \end{bmatrix}_0$$

and the initial estimate of the control program $u(t)$.

The performance index to be minimized is

$$\phi = t_f$$

and the terminal constraints are

$$\Psi_1 = u(t_f) - u_f = 0$$

$$\Psi_2 = v(t_f) - v_f = 0$$

$$\Psi_3 = r(t_f) - r_f = 0$$

The condition that is used to stop the integration is

$$\Omega = \theta(t_f) - \theta_f = 0 .$$

The equations adjoint to the differential equations of motion, $\dot{\lambda} = -f_x^T \lambda$, are

$$\dot{\lambda}_1 = \left(\frac{v}{r}\right) \lambda_2 - \lambda_3$$

$$\dot{\lambda}_2 = -\left(\frac{2v}{r}\right) \lambda_1 + \left(\frac{u}{r}\right) \lambda_2 - \left(\frac{1}{r}\right) \lambda_4$$

$$\dot{\lambda}_3 = -\left(\frac{2GM}{r^3} - \frac{v^2}{r^2}\right) \lambda_1 - \left(\frac{uv}{r^2}\right) \lambda_2 + \left(\frac{v}{r^2}\right) \lambda_4$$

$$\dot{\lambda}_4 = 0 \ ,$$

and the starting conditions for the backward integration are,

$$\lambda_\phi^T(t_f) = \left[\frac{\partial \phi}{\partial x}\right]_f = [0 \quad 0 \quad 0 \quad 0]$$

$$\lambda_\psi^T(t_f) = \left[\frac{\partial \psi}{\partial x}\right]_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\lambda_\Omega^T(t_f) = \left[\frac{\partial \Omega}{\partial x}\right]_f = [0 \quad 0 \quad 0 \quad 1] \ .$$

The time rates of change of the performance index, terminal constraints and stopping condition are

$$\dot{\phi} = \left[\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \dot{x}\right]_f = 1$$

$$\dot{\psi} = \left[\frac{\partial \psi}{\partial t} + \frac{\partial \psi}{\partial x}\dot{x}\right]_f = \begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{r}\end{bmatrix}_f$$

$$\dot{\Omega} = \left[\frac{\partial \Omega}{\partial t} + \frac{\partial \Omega}{\partial x}\dot{x}\right]_f = \dot{\theta}_f \ .$$

The starting conditions for $I_{\psi\psi}$, $I_{\psi\phi}$ and $I_{\phi\phi}$ are

$$I_{\psi\psi}(t_f) = \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$$

$$I_{\psi\phi}(t_f) = \begin{bmatrix}0\\ 0\\ 0\end{bmatrix}$$

$$I_{\phi\phi}(t_f) = 0 \ .$$

APPENDIX A.2.5

Application of the Modified Method of Steepest Descent

The nonlinear, ordinary, vector differential equation $\dot{x} = f(x,u,t)$ is composed of $n = 4$ differential equations of motion. These equations are integrated forward with the initial conditions

$$x(t_0) = \begin{bmatrix} u \\ v \\ r \\ \theta \end{bmatrix}_0$$

and the initial estimate of the control program $u(t)$.

The penalty function to be minimized is

$$P = W_0 t_f^2 + W_1[u(t_f)-u_f]^2 + W_2[v(t_f)-v_f]^2 + W_3[r(t_f)-r_f]^2$$

and the stopping condition is

$$\Omega = \theta(t_f) - \theta_f = 0 .$$

The equations adjoint to the differential equations of motion, $\dot{\lambda} = -f_x{}^T \lambda$, are

$$\dot{\lambda}_1 = \left(\frac{v}{r}\right)\lambda_2 - \lambda_3$$

$$\dot{\lambda}_2 = -\left(\frac{2v}{r}\right)\lambda_1 + \left(\frac{u}{r}\right)\lambda_2 - \left(\frac{1}{r}\right)\lambda_4$$

$$\dot{\lambda}_3 = -\left(\frac{2GM}{r^3} - \frac{v^2}{r^3}\right)\lambda_1 - \left(\frac{uv}{r^2}\right)\lambda_2 + \left(\frac{v}{r^2}\right)\lambda_4$$

$$\dot{\lambda}_4 = 0 \ .$$

The starting conditions for the backward integration are

$$\lambda^T_{P\Omega}(t_f) = \left\{\left[\frac{\partial P}{\partial x}\right] - \left[\frac{J_P}{J_\Omega}\right]\left[\frac{\partial \Omega}{\partial x}\right]\right\} = [\lambda_{P\Omega_1}, \lambda_{P\Omega_2}, \lambda_{P\Omega_3}, \lambda_{P\Omega_4}]$$

where

$$\lambda_{P\Omega_1} = 2W_1[u(t_f) - u_f]$$

$$\lambda_{P\Omega_2} = 2W_2[v(t_f) - v_f]$$

$$\lambda_{P\Omega_3} = 2W_3[r(t_f) - r_f]$$

$$\lambda_{P\Omega_4} - \left[\frac{\lambda_{P\Omega_1} f_1 + \lambda_{P\Omega_2} f_2 + \lambda_{P\Omega_3} f_3 + 2W_0 t_f}{f_4}\right] .$$

The new control program is given by

$$\delta u = K\lambda^T_{P\Omega}G = K\left[\frac{T}{m}(\lambda_{P\Omega_1} \sin\beta - \lambda_{P\Omega_2} \cos\beta)\right] .$$

APPENDIX A.3

Numerical Constants

Earth-Mars Transfer.

Astronomical Unit, AU	$.14959870 \times 10^{12}$ meters
Orbital Radius of Earth, r_e	$.10000000 \times 10^{1}$ AU
Orbital Radius of Mars, r_m	$.15236790 \times 10^{1}$ AU
Gravitational Constant of Sun, GM_s	$.13271504 \times 10^{21}$ meters^3/second^2
Initial Spacecraft Mass, m_o	$.67978852 \times 10^{3}$ kilograms
Thrust, T	$.40312370 \times 10^{1}$ newtons
Mass Rate, $\dot{m}$	$.10123858 \times 10^{-4}$ kilograms/second

Earth Launch

Radius of Earth, R_e	$.63781700 \times 10^{7}$ meters
Gravitational Constant of Earth, GM_e	$.39860640 \times 10^{15}$ meters^3/second^2
Initial Spacecraft Mass, m_o	$.15000000 \times 10^{4}$ kilograms
Thrust, T	$.27000000 \times 10^{5}$ newtons
Mass Rate, $\dot{m}$	$.45000000 \times 10^{1}$ kilograms/second

The terms that must be added to the differential equations f_1 and f_2 to include atmospheric resistance are:

$$f_1 = f_1 - \frac{\rho C_D A u \sqrt{u^2+v^2}}{2m}$$

$$f_2 = f_2 - \frac{\rho C_D A v \sqrt{u^2+v^2}}{2m}$$

where $\rho = \rho_0 e^{-\frac{r-R_e}{E}}$ (mass density)

$C_D = 0.3, \ 0 < M < .6950$ (drag coefficient)

$C_D = K_1 + \frac{K_2}{M^2} - \frac{K_3}{M^3}, \ M > .6950$

$M = \frac{\sqrt{u^2+v^2}}{a}$ (mach number)

$a = D - B(r - R_e)$ (speed of sound)

and where

$\rho_0 = 0.52$ kilograms/meter^3

$E = 7600.0$ meters

$K_1 = 0.1368$

$K_2 = 1.6218$

$K_3 = 1.0724$

$D = 340.0$ meters/second

$B = 0.00071$ ^{1}/seconds

$A = 4.0$ meters^2

APPENDIX A.4

Normalization Scheme

Earth-Mars Transfer

Unit of Length (1 AU)	$.14959870 \times 10^{12}$ meters
Unit of Mass (m_o)	$.67978852 \times 10^{3}$ kilograms
Unit of Velocity $v_e = \sqrt{\frac{GM_s}{r_e}}$	$.29784901 \times 10^{5}$ meters/second
Unit of Force	$.40312370 \times 10^{1}$ newtons
Unit of Time	$.50226355 \times 10^{7}$ seconds
	$.58132355 \times 10^{2}$ days

The normalized values of the parameters of interest are:

Gravitational Constant of Sun, $GM_s = 1.0$

Initial Spacecraft Mass = 1.0

Initial Spacecraft Velocity = 1.0

Initial Spacecraft Radius = 1.0

Terminal Spacecraft Velocity = 0.81012728

Terminal Spacecraft Radius = 1.5236790

Thrust = .14012969

Mass Rate = 0.074800391

Earth Launch - No normalization scheme.

BIBLIOGRAPHY

1. Lawden, D. F., "Optimal Programming of Rocket Thrust Direction," presented at the Fifth I.A.F. Congress at Innsbruck, August 6, 1954.

2. Breakwell, J. V., "The Optimization of Trajectories," North American Aviation, Inc., Report AL-2706, August 1957.

3. Kelley, H. J., "Gradient Theory of Optimal Flight Paths," ARS Journal (now AIAA Journal), p. 947, October 1960.

4. MacKay, J. S., Rossa, L. G., and Zimmerman, A. V., "Optimum Low-Acceleration Trajectories for Earth-Mars Transfer," Proceedings of the IAS Symposium on Vehicle System Optimization, Garden City, N.Y., Nov. 28-29, 1961.

5. Melbourne, W. G., Sauer, C. G., Jr., and Richardson, D. E., "Interplanetary Trajectory Optimization with Power-Limited Propulsion Systems," Proceedings of the IAS Symposium on Vehicle System Optimization. Garden City, N.Y., Nov. 28-29, 1961.

6. Scharmack, D. K., "The Equivalent Minimization Problem and the Newton-Raphson Optimization Method," Proceedings from Optimum System Synthesis Conference, Wright Patterson Air Force Base, September 1962, p. 119-158.

7. Jurovics, S. A. and McIntyre, J. E., "The Adjoint Method and its Application to Trajectory Optimization," ARS Journal, p. 1354, September 1962.

8. Goodman, T. R. and Lance, G. N., "The Numerical Integration of Two-Point Boundary Value Problems," Math Tables and Other Aids to Computation, Vol. 10, p. 82, 1956.

9. Breakwell, J. V., Speyer, J. L., and Bryson, A. E., "Optimization and Control of Nonlinear Systems Using the Second Variation," Journal of the Society of Industrial and Applied Mathematics Control, Ser. A, Vol. 1, No. 2, p. 193-223, 1963.

10. Breakwell, J. V., "The Optimization of Trajectories," Journal of the Society of Industrial and Applied Mathematics, Vol. 7, No. 2, p. 215-247, June 1959.

11. Kelley, H. J., Kopp, R. E., and Moyer, H. G., "A Trajectory Optimization Technique Based Upon the Theory of the Second Variation," presented at the AIAA Astrodynamics Conference, Yale University, New Haven, Conn., August 19-21, 1963.

12. Jazwinski, A. H., "Optimal Trajectories and Linear Control of Nonlinear Systems," AIAA Journal, Vol. 2, No. 8, p. 1371, August 1964.

13. McGill, R. and Kenneth, P., "Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator," AIAA Journal, Vol. 2, No. 10, p. 1761, October 1964.

14. Kalaba, R., "Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence," Journal of Mathematics and Mechanics, Vol. 8, No. 4, p. 519-574, 1959.

15. McGill, R. and Kenneth, P., "A Convergence Theorem on the Iterative Solution of Nonlinear Two-Point Boundary Value Systems," presented at the XIV International Astronautical Congress, Paris, September 25, 1963.

16. Long, R. S., "Newton-Raphson Operator; Problems with Undetermined End Points," AIAA Journal, Vol. 3, No. 7, p. 1352, July 1965.

17. Bryson, A. E., Denham, W. F., Carroll, F. J., and Mikami, K., "Determination of Lift or Drag Programs to Minimize Re-Entry Heating," Journal of the Aerospace Sciences, p. 420, April 1962.

18. Bryson, A. E. and Denham, W. F., "A Steepest-Ascent Method for Solving Optimal Programming Problems," Journal of Applied Mechanics, p. 247, June 1962.

19. Kelley, H. J., Kopp, R. E., and Moyer, H. G., "Successive Approximation Techniques for Trajectory Optimization," Proceedings of the IAS Symposium on Vehicle System Optimization, Garden City, N.Y., Nov. 28-29, 1961.

20. Pfeiffer, C. G., "Theory and Application of the Critical Direction Method of Trajectory Optimization," Proceedings of the IAS Symposium on Vehicle System Optimization, Garden City, N.Y., Nov. 28-29, 1961.

21. Wagner, W. E., Jazwinski, A. H., "Three-Dimensional Re-Entry Optimization with Inequality Constraints," For Presentations, Astrodynamics Conference of the AIAA, Yale University, New Haven, Conn., August 19-21, 1963.

22. Rosenbaum, R., "Convergence Technique for the Steepest-Descent Method of Trajectory Optimization," AIAA Journal, Vol. 1, No. 7, p. 1704, July 1963.

23. Stancil, R. T., "A New Approach to Steepest-Ascent Trajectory Optimization," AIAA Journal, Vol. 2, No. 8, p. 1365, August 1964.

24. McReynolds, S. R. and Bryson, A. E., Jr., "A Successive Sweep Method for Solving Optimal Programming Problems," Tech. Report No. 463, Cruft Laboratory, Harvard University, Cambridge, Mass., March 1965.

25. Kopp, R. E. and McGill, R., "Several Trajectory Optimization Techniques Part I," Computing Methods in Optimization Problems, Academic Press, 1964.

26. Moyer, Gardner H. and Pinkham, Gordon, "Several Trajectory Optimization Techniques Part II," Computing Methods in Optimization Problems, Academic Press, 1964.

27. Gelfand, I. M. and Fomin, S. V., Calculus of Variations, Prentice Hall, 1963.

28. Melbourne, W. G., "Three-Dimensional Optimum Thrust Trajectories for Power Limited Propulsion Systems," ARS Journal, December 1961, p. 1723.

29. Merriam, C. W., "An Algorithm for the Iterative Solution of a Class of Two-Point Boundary Value Problems, Journal of the Society of Industrial and Applied Mathematics Control, Ser. A., Vol. 2, No. 1, pp. 1-10, 1964.

30. Sylvester, R. J. and Meyer, F., "Two Point Boundary Problems by Quasilinearization," Journal of the Society of Industrial and Applied Mathematics, Vol. 13, No. 2, pp. 586-602, June 1965.